



UNIVERSITAT DE
BARCELONA

Treball final de grau

GRAU D'ENGINYERIA
INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

ACTUALITZACIÓ I MILLORA
DE LA GUIA WEB PER A L'ÚS
DE MEDICAMENTS EN
INSUFICIÈNCIA RENAL

Autor: Aleix Marqués Casasnovas

Director: Laura Igual Muñoz
Realitzat a: Departament de
Matemàtiques i Informàtica

Barcelona, 27 de juny de 2018

Resum

En aquest projecte es treballa en actualitzar i ampliar la guia web per a l'ús de medicaments en insuficiència renal desenvolupada en la Facultat de Matemàtiques i Informàtica en col·laboració amb l'equip de Farmàcia Pràctica. L'objectiu principal és permetre guardar dades sobre l'ús d'aquesta guia web. Amb aquestes dades les facultats de Matemàtiques i Informàtica, i de Farmàcia podrà realitzar un estudi per a poder descobrir quins medicaments són més receptats a persones que, degut a l'estat del seu ronyó, podrien tenir problemes secundaris perjudicials si prenen el medicament de la mateixa forma que un pacient sa. L'altre objectiu d'aquesta millora es augmentar la usabilitat i adaptabilitat de la pàgina per a facilitar l'ús de la mateixa en dispositius mòbils. Per tal d'aconseguir aquests objectius es canvien les tecnologies que es van utilitzar durant la creació de la pàgina per altres més actuals i que ofereixen un millor rendiment. Per últim s'amplia la web i s'afegeix una nova funcionalitat a la pàgina que permet recollir dades sobre els pacients, els problemes de salut que pateixen i quins medicaments utilitzen per a tractar-los.

Resumen

En este proyecto se trabaja en actualizar y ampliar la guía web para el uso de medicamentos en insuficiencia renal desarrollada en la Facultad de Matemáticas e Informática en colaboración con el equipo de Farmacia Práctica. El objetivo principal es permitir guardar datos sobre el uso de esta guía web. Con estos datos las facultades de Matemáticas e Informática, y de Farmacia podrá realizar un estudio para poder descubrir qué medicamentos son más recetados a personas que, debido al estado de su riñón, podrían tener problemas secundarios perjudiciales si toman el medicamento de la misma forma que un paciente sano. El otro objetivo de esta mejora es aumentar la usabilidad y adaptabilidad de la página para facilitar el uso de la misma en dispositivos móviles. Para conseguir estos objetivos se cambian las tecnologías que se utilizaron durante la creación de la página por otros más actuales y que ofrecen un mejor rendimiento. Por último se amplía la web y se añade una nueva funcionalidad a la página que permite recoger datos sobre los pacientes, los problemas de salud que sufren y qué medicamentos utilizan para tratarlos.

Abstract

This project goal is updating and extending the web guide for the use of renal insufficiency medications developed in the Faculty of Mathematics and Computing in collaboration with the Pharmacy Practice team. The main objective is to allow to save data on the use of this web guide. With this data the faculties of Mathematics and Computing, and the Faculty of Pharmacy will be able to conduct a study to discover which medications are most prescribed for people who, due to the state of their kidney, may have harmful secondary problems if they take the medication like any other healthy patient. The other objective of this improvement is to increase the usability and responsive of the page to facilitate its use on mobile devices. In order to achieve these goals, the technologies that were used during the creation of the page are changed for more current ones that offer a better performance. Finally, the web is expanded and a new feature is also added to the page that allows you to collect data about patients, the health problems they suffer, and

what medications they use to treat them.

Índex

1	Introducció	1
1.1	El projecte	1
1.2	El desenvolupament del projecte	1
2	Glossari	5
3	Objectius	6
4	Anàlisi	8
4.1	Anàlisi de Requeriments i Funcionalitats	8
4.2	Requeriments funcionals	8
4.3	Casos d'ús	11
4.4	Requeriments no funcionals	13
5	Planificació	16
5.1	Planificació inicial	16
5.2	Compliment de la planificació	16
6	Disseny	18
6.1	Definició del model	18
6.2	Model Entitat-Relació	19
6.3	Definició del model relacional	20
7	Tecnologies utilitzades	23
7.1	Servidor Web	23
7.2	Lumen	24
7.3	Composer	25
7.4	Bootstrap	25
7.5	Angular	25
7.6	NodeJs i npm	27
8	Desenvolupament	28
8.1	Aplicació Angular	28
8.2	Aplicació Lumen	38
8.3	Integració de les dues parts	44

8.4	Pàgina de pacients	44
8.5	Part de client	47
9	Proves i resultats	50
9.1	Pantalles	50
10	Conclusions i treball futur	55
10.1	Conclusions	55
10.2	Treball Futur	55
	Referències	56
11	Apèndix	59
11.1	Guia d'instal·lació del servidor local	59

1 Introducció

1.1 El projecte

Aquest projecte consisteix en l'actualització de la guia web de medicaments en insuficiència renal elaborat per un equip de la Facultat de Matemàtiques i Informàtica a petició de la Facultat de Farmàcia [1].

La idea d'aquesta web sorgeix del Grup assistencial de docència i investigació en l'àmbit de Farmàcia Pràctica de la Universitat de Barcelona. Aquest Grup està integrat per professors de la UB i professionals farmacèutics. Els membres del Grup comparteixen l'objectiu de desenvolupar i millorar els serveis d'atenció farmacèutica mitjançant la generació de coneixement sobre l'ús òptim, segur i efectiu dels medicaments per part dels pacients. A més, desenvolupen eines per facilitar la transferència d'aquest coneixement a la pràctica professional per així contribuir decisivament a la millora del benestar i la salut dels ciutadans [2].

L'estudi previ al desenvolupament de la web consisteix en crear una base de dades de possibles efectes secundaris dels diferents principis actius dels medicaments en funció del Filtrat Glomerular estimat (FGe), i ha estat realitzat per les Doctores Marian March Pujol, M. Angels Via-Sosa i Natali Lopes, membres del Grup de Farmàcia Pràctica. Aquest estudi demostra que un de cada tres pacients polimedicats majors de 65 anys que acudeix a les farmàcies per a que se li dispensin els medicaments prescrits presenta un deteriorament de la funció renal i almenys un d'aquests medicaments requereix d'un ajust posològic per a poder garantir la seguretat del medicament per a aquest pacient [3].

L'altra iniciativa és el Conveni de Col·laboració entre la Universitat de Barcelona i la Societat Espanyola de Nefrologia promogut per la Facultat de Farmàcia el 28 de desembre de 2013 [4]. L'objectiu d'aquest conveni és promocionar la col·laboració, formació i comunicació entre metges i farmacèutics. Arrel d'aquest conveni es crea un grup de treball integrat per membres de la S.E.N. i del Grup per revisar i consensuar la guia de medicaments que requereixen un ajust posològic en pacients amb una funció renal disminuïda. S'han analitzat els medicaments més distribuïts a les farmàcies comunitàries segons la classificació ATC [5] i s'indiquen els símptomes per sobredosis així com els nivells de risc que suposaria per a un pacient prendre un medicament determinat en funció del seu nivell de FGe.

1.2 El desenvolupament del projecte

À l'any 2015, es decideix la creació d'una plana web [1] per a centralitzar, compartir i consultar de manera telemàtica aquesta base de dades. El desenvolupament d'aquesta web s'ha realitzat en tres fases. La primera fase consisteix en la creació

d'una pàgina web que serveixi com a interfície gràfica per a consultar la guia de medicaments [6]. Durant la segona fase s'implementen noves funcionalitats per a interactuar amb la base de dades [7]. La tercera fase de desenvolupament es la que es realitza en aquest projecte.

Primera fase de desenvolupament

Durant aquesta fase s'ha dissenyat i desenvolupat un sistema de gestió telemàtica mitjançant una plana web que permetí cercar i interactuar de forma més intuïtiva els medicaments de la base de dades. L'objectiu principal d'aquesta fase és, per tant, implementar una funcionalitat de cerca que recuperi de la base de dades tota la informació d'un medicament a partir del seu nom. Per aconseguir aquest objectiu de forma satisfactòria s'han analitzat les necessitats funcionals de la Facultat de Farmàcia per a trobar la forma òptima de gestionar el contingut de la guia. Durant aquesta fase s'ha dissenyat la web com un sistema tancant on només hi tindrien accés alumnes, doctors i farmacèutics.

Segona fase de desenvolupament

Durant la segona fase del projecte l'objectiu és millorar la forma de gestionar la base de dades de la web. Per aconseguir aquest objectiu es decideix la implementació de noves funcionalitats CRUD (Create, Recover, Update, Delete), disponibles per a l'usuari administrador de la web, per als principis actius i les seves recomanacions. A més es renova l'aspecte gràfic i s'obre la web al públic. És a dir, es permet que qualsevol persona es pugui registrar a la web.

Limitacions del projecte a l'inici de la tercera fase

A l'inici d'aquest treball la pàgina web està programada amb AngularJs per al frontend i Laravel 4 per al backend. La pàgina té problemes d'usabilitat i adaptabilitat. Principalment degut a que falten missatges d'informació i que s'està abusant dels modals. Això provoca que en dispositius mòbils es detecti una pobre experiència d'usuari. Un altre problema es que no s'estan guardant les cerques fetes a la base de dades pel qual no es pot fer cap tipus d'anàlisi a partir d'aquesta informació, que com s'explicarà més endavant, es de gran importància per a l'equip de la Facultat de Farmàcia.

Hi ha pàgines com la d'usuaris que tarden més de deu segons en carregar ja que quan es van desenvolupar no és va tenir en compte l'impacte que tindria en el rendiment el fet de recuperar tota la taula d'usuaris quan aquesta contingues una

gran quantitat de dades.

Hi ha altres limitacions que no es detallen en aquest apartat però que s'entendran dels objectius secundaris enumerats al tercer capítol d'aquesta memòria. No hi ha cap guia per a configurar l'entorn local per a començar a desenvolupar. Simplement muntant un servidor Linux amb l'*stack* LAMP no serveix, ja que es necessiten versions antigues tant de Laravel com de PHP. També es necessari tenir certs mòduls activats en el servidor Apache. Això, que per un desenvolupador amb experiència pot semblar trivial, quan no se'n té no resulta tant evident que s'ha de fer per a solucionar els errors i poder començar el desenvolupament.

Finalment, no hi ha cap servei d'emmagatzematge de les cerques realitzades en la web. Aquest fet provoca que les cerques realitzades des de la primera fase, fa ja quasi tres anys, no es puguin utilitzar al no estar registrades en cap lloc. L'únic que es té es el registre d'accés que genera automàticament el servei d'hostatgeria web la Universitat de Barcelona, però, degut al plantejament de la REST Api, no proporcionen la informació necessària per a poder realitzar un projecte d'anàlisi de dades. Projecte, que com es comenta més endavant, té bastant interès per a les Facultats de Matemàtiques i Informàtica i de Farmàcia.

Tercera fase de desenvolupament

En aquesta fase l'objectiu principal és preparar la web per a poder recollir dades estructurades sobre l'ús de la guia que permetin, a la següent fase de desenvolupament, poder fer un anàlisi estadístic. L'objectiu secundari és millorar la usabilitat i adaptabilitat de la pàgina, així com donar solució a la resta de limitacions esmentades anteriorment.

Per tal d'aconseguir l'objectiu principal s'han de guardar totes les cerques que es realitzin a la guia i implementar una nova funcionalitat que permeti la introducció de dades de pacients i els medicaments que prenen per mitjà d'un formulari basat en unes taules proporcionades pel propi equip de Farmàcia. Aquestes taules són les que utilitzen els alumnes de darrer curs de farmàcia a les seves pràctiques externes i, per tant, es preveu poder recollir dades de més d'un centenar de pacients només contant amb les dades que introdueixin els alumnes. A més, ja que els alumnes cada any entreguen aquestes taules en paper, es podrà traspasar aquesta informació a la base de dades utilitzant aquesta nova funcionalitat i així poder disposar de una major quantitat de dades per a poder analitzar.

Quarta fase de desenvolupament

En aquesta fase es on es realitzarà un projecte de ciència de dades a partir de les dades recollides com a resultat de la feina realitzada en aquest projecte. L'objectiu es descobrir tendències a partir dels medicaments més consultats a la guia i les dades del seguiment dels diferents pacients. Des de Farmàcia estan molt interessats en saber quins son els medicaments més consultats i, per tant, més receptats a les farmàcies. Dades que s'obtindran a partir dels registres de cerques a la guia. També, a partir de les dades recollides amb la nova funcionalitat de pacients preveuen poder aconseguir relacions; com saber quins medicaments son més receptats als pacients segons la seva edat o quins medicaments son més perjudicials en funció del nivell de FGe del pacient o la seva edat.

2 Glossari

- **Filtratge Glomerular estimat (FGe):**

És una mesura del funcionalisme dels ronyons. Aquesta prova utilitza el resultat de la determinació de creatinina en sang, incloent-lo en una fórmula d'estimació del filtrat glomerular, el resultat reflecteix el grau de funcionament dels ronyons.

- **Suggestiment:**

Missatge enviat per un usuari a l'administrador de la pàgina on pot comentar una millora que poder realitzar a la web o un defecte que hagi trobat i consideri que s'ha de solucionar. És pot inclou-re un principi actiu de la guia per a comentar algun canvi que sigui necessari aplicar a les dades guardades sobre dit principi actiu.

- **Recomanació:**

Conjunt d'informació sobre l'efecte d'un principi actiu en un pacient que tingui un nivell de FGe en un rang determinat. S'indica el nivell de risc, l'ajust de la dosi i el seguiment a realitzar al pacient de ser necessaris.

- **RNM:**

Resultat Negatiu associat a la Medicació

- **PRM:**

Problema Relacionat amb el Medicament

- **Intervenció:**

Acció a realitzar sobre un PRM per a evitar un efecte negatiu en la salut del pacient.

3 Objectius

Els dos objectius principals d'aquest projecte són:

1. Recopilació de dades estructurades sobre l'ús de la guia i sobre pacients i medicaments.
2. Millora de la usabilitat i adaptabilitat de la pàgina.

Aquests dos objectius es poden desglossar en requisits concrets. Per al primer objectiu principal s'ha de complir:

- 1.1 Guardar les cerques realitzades a la guia web.
- 1.2 Implementar una pàgina que permeti introduir dades sobre pacients i els medicaments que prenen, que com ja s'ha comentat a l'apartat 1.2 són de gran interès per a l'equip de farmàcia.

Per al segon objectiu, s'han de complir els següents requisits:

- 2.1 Actualitzar les tecnologies que utilitza la pàgina web abans de començar la tercera fase. En concret s'actualitza la versió d'Angular, de la 1 a la darrera, la 6; i la versió de Bootstrap, de la 3 a la 5. També es canvia la tecnologia del servidor, de Laravel 4 a Lumen 5 i de PHP 5 a PHP 7. Els motius d'aquests canvis s'expliquen més detall a la secció 5.
- 2.2 Afegir missatges d'informació a la pàgina que permetin a l'usuari saber el resultat de les seves accions.
- 2.3 Eliminar els pop-ups i modals que perjudiquin l'experiència d'usuari en dispositius mòbils.
- 2.4 Afegir adaptabilitat a les taules de la web.
- 2.5 Afegir validacions, tant a la part de client com a la part de servidor, als diferents formularis de la guia per assegurar la integritat i consistència de les dades.

A més, aquest projecte té aquests altres objectius secundaris:

3. Permetre exportar la base de dades a fitxers CSV.
4. Permetre que l'administrador pugui eliminar qualsevol usuari, a excepció d'ell mateix.

5. Pujar el codi font a un repositori privat de la UB per a que estigui disponible per a la quarta fase de desenvolupament. Aquest punt es bastant important ja que les noves version d'Angular han de ser compilades i per tant els fitxers que s'allotjaran al servidor no podran utilitzar-se per a seguir desenvolupant.
6. Crear una guia d'instal·lació per al servidor local per tal de reduir el temps d'entrada al projecte, és a dir, reduir el temps necessari de recopilació d'informació per a desplegar l'estructura necessària per a començar a desenvolupar.
7. Crear una guia que expliqui els passos a seguir des de que s'acaba el desenvolupament fins que es puja al servei d'hostatgeria web de la UB. Aquesta guia es bastant important ja que el servei d'hostatgeria web de la UB no disposa de Composer, un dels principal gestors de dependències de PHP, instal·lat als seus servidor. Per tant es necessari que el desenvolupador instal·li la mateixa versió de PHP que hi ha al servidors del servei web de la UB en la seva màquina per a descarregar-se les versions correctes de les dependències abans de poder fer la pujada [8].
8. Afegir **Google Analytics** a la pàgina per tal de poder reunir dades sobre l'ús de la pàgina.

4 Anàlisi

4.1 Anàlisi de Requeriments i Funcionalitats

Primer es descriuran les funcions específiques que s'han d'implementar per mitjà de requeriments funcionals. Tot seguit es mostraran els requeriments de comportament mitjançant diagrames de casos d'ús. Finalment s'explicaran les característiques que han de complir tots els requisits funcionals per tal d'assegurar un rendiment i experiència d'usuari òptima al mateix temps que s'assegura la màxima seguretat i consistència de les dades.

4.2 Requeriments funcionals

Primer es mostraran les funcionalitats que ja es troben implementades a l'inici de la tercera fase. A continuació es descriuran les noves funcionalitats que ha de tenir disponibles tots els usuaris. Tot seguit s'explicaran les funcionalitats afegides a l'usuari registrat i finalment les que s'han afegir a l'usuari administrador.

Funcionalitats ja implementades

1. Registrar-se a la pàgina.
2. Rebre un correu per activar la seva compta.
3. Autenticar-se a la pàgina, primer ha d'activar la compta.
4. Poder recuperar la seva contrasenya en cas que l'oblidi.
5. Poder cercar informació sobre un principi actiu a partir del nom.
6. Poder filtrar la informació sobre un principi actiu en funció del FGe.
7. Poder veure la informació de la compta d'usuari.
8. Poder canviar la contrasenya.
9. Poder donar de baixa la compta.
10. Poder enviar un suggeriment a l'administrador.
11. Poder tancar la seva sessió.
12. Poder eliminar tots els suggeriments.
13. Poder eliminar tots els usuaris.
14. Poder pujar un fitxer Excel amb tots els principis actius.

15. Poder eliminar tots els principis actius.
16. Poder pujar un fitxer Excel amb totes les recomanacions dels principis actius.
17. Poder eliminar tots els principis actius.
18. Poder afegir un nou principi actiu.
19. Poder eliminar un principi actiu determinat.
20. Poder actualitzar la informació d'un principi actiu determinat.
21. Poder afegir una nova recomanació sobre un principi actiu determinat.
22. Poder eliminar una recomanació determinada.
23. Poder actualitzar la informació sobre una recomanació determinada.
24. Poder visualitzar tots els suggeriments.
25. Poder visualitzar totes els usuaris.

D'aquestes funcionalitats n'hi ha algunes que no s'han de mantenir a la tercera fase ja que s'ha considerat que han deixat de ser necessàries. Aquestes funcionalitats no necessàries són les funcionalitats de la 11 a la 16, ambdues incloses, i que, per tant, no es mantenen en aquest projecte. A continuació s'agrupen les funcionalitats anteriors en funció del tipus d'usuari que hi té accés i a més s'indiquen les funcionalitats noves a implementar durant aquest projecte.

Funcionalitats per a tots els usuaris

A més de les funcionalitats 1 a la 3 anteriors, s'han d'implementar dues funcionalitats més.

26. Saber per a que serveix i que aporta la guia web sense necessitar registrar-se primer.
27. Poder redimensionar la pàgina sense que afecti negativament a l'experiència d'usuari.

Funcionalitats de l'usuari registrat

L'usuari registrat té accés a les funcionalitats compreses entre la 4 i la 11. A més es necessari implementar les següents funcionalitats durant la tercera fase.

28. Poder enviar un suggeriments sense adjuntar un principi actiu.

29. Poder veure el llistat de pacients que ha introduït.
30. Poder veure les dades d'un pacient que hagi introduït.
31. Poder introduir les dades d'un pacient.
32. Poder editar les dades d'un pacient.
33. Poder eliminar les dades d'un pacient.
34. Poder introduir les dades d'un problema de salut d'un pacient concret.
35. Poder editar un problema de salut d'un pacient concret.
36. Poder eliminar un problema de salut d'un pacient concret.
37. Poder introduir un tractament relacionat amb un problema de salut d'un pacient concret.
38. Poder editar les dades d'un tractament relacionat amb un problema de salut d'un pacient concret.
39. Poder eliminar un tractament relacionat amb un problema de salut d'un pacient concret.
40. Poder introduir un RNM d'un problema de salut d'un pacient concret.
41. Poder editar les dades d'un RNM d'un problema de salut d'un pacient concret.
42. Poder eliminar un RNM d'un problema de salut d'un pacient concret.
43. Poder introduir les dades d'un PRM relacionat amb un RNM i un tractament d'un problema de salut d'un pacient concret.
44. Poder editar les dades d'un PRM relacionat amb un RNM i un tractament d'un problema de salut d'un pacient concret.
45. Poder eliminar les dades d'un PRM relacionat amb un RNM i un tractament d'un problema de salut d'un pacient concret.
46. Per a cada PRM poder seleccionar una o més intervencions a realitzar.

Noves funcionalitats de l'usuari administrador

L'usuari administrador té accés a les funcionalitats compreses entre la 11 i la 25. A més s'han d'implementar les següents funcionalitats.

47. Poder eliminar un usuari en concret sempre que aquest usuari no tingui el rol d'administrador.
48. Poder veure la data d'enviament d'un suggeriment.

49. Poder veure si la base de dades conté principis actius duplicats.
50. Poder veure una comparativa de les dades contingudes en principis actius duplicats per a poder eliminar un dels dos.
51. Poder veure si la base de dades conté principis actius amb recomanacions que tinguin rangs de risc superposats.
52. Poder veure les recomanacions dels principis actius que tenen els rangs de risc superposats per a poder corregir-los.

4.3 Casos d'ús

A continuació es mostren els diferents casos d'ús que s'han de garantir al finalitzar aquesta tercera etapa de desenvolupament.

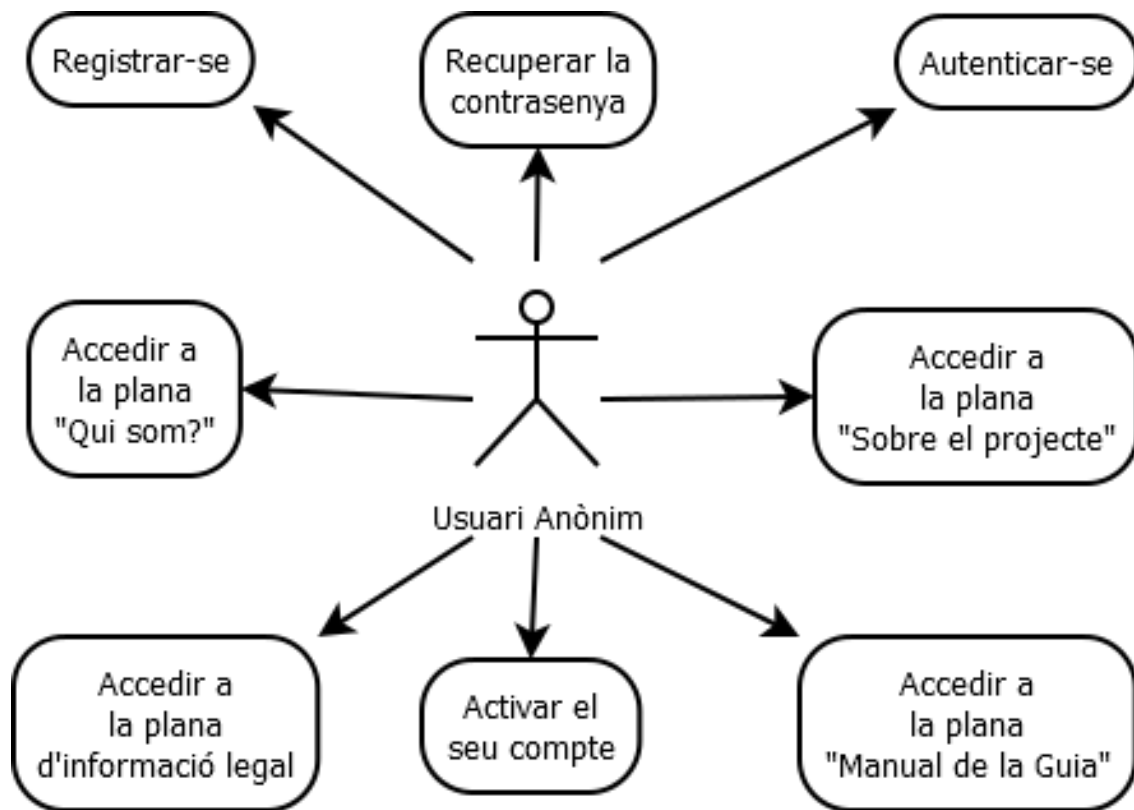


Figura 1: Casos d'ús de l'usuari anònim

Es considera l'usuari anònim com aquell que encara no ha iniciat sessió a la plana web.

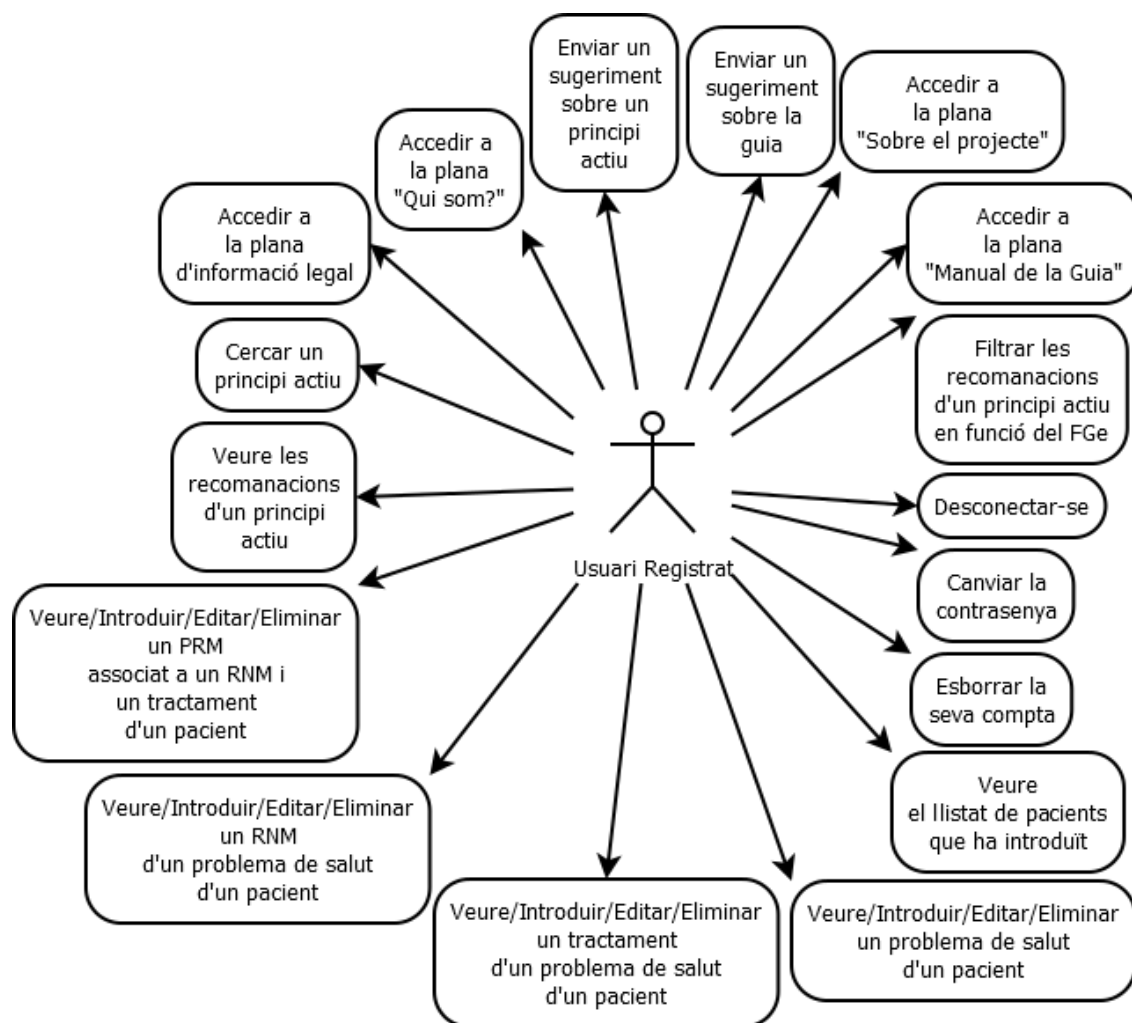


Figura 2: Casos d'ús de l'usuari registrat

Es considera un usuari registrat a tot aquell usuari anònim que s'ha registrat mitjançant el formulari de registre de la pàgina web i ha activat la compta amb l'enllaç que se l'hi ha enviat per correu.

Els nous casos d'ús de l'usuari registrat referents a pacients són els que fan necessari la creació de una nova pàgina per a poder veure i introduir aquestes noves dades. Aquesta pàgina s'explica més en detall a l'apartat 8.4.

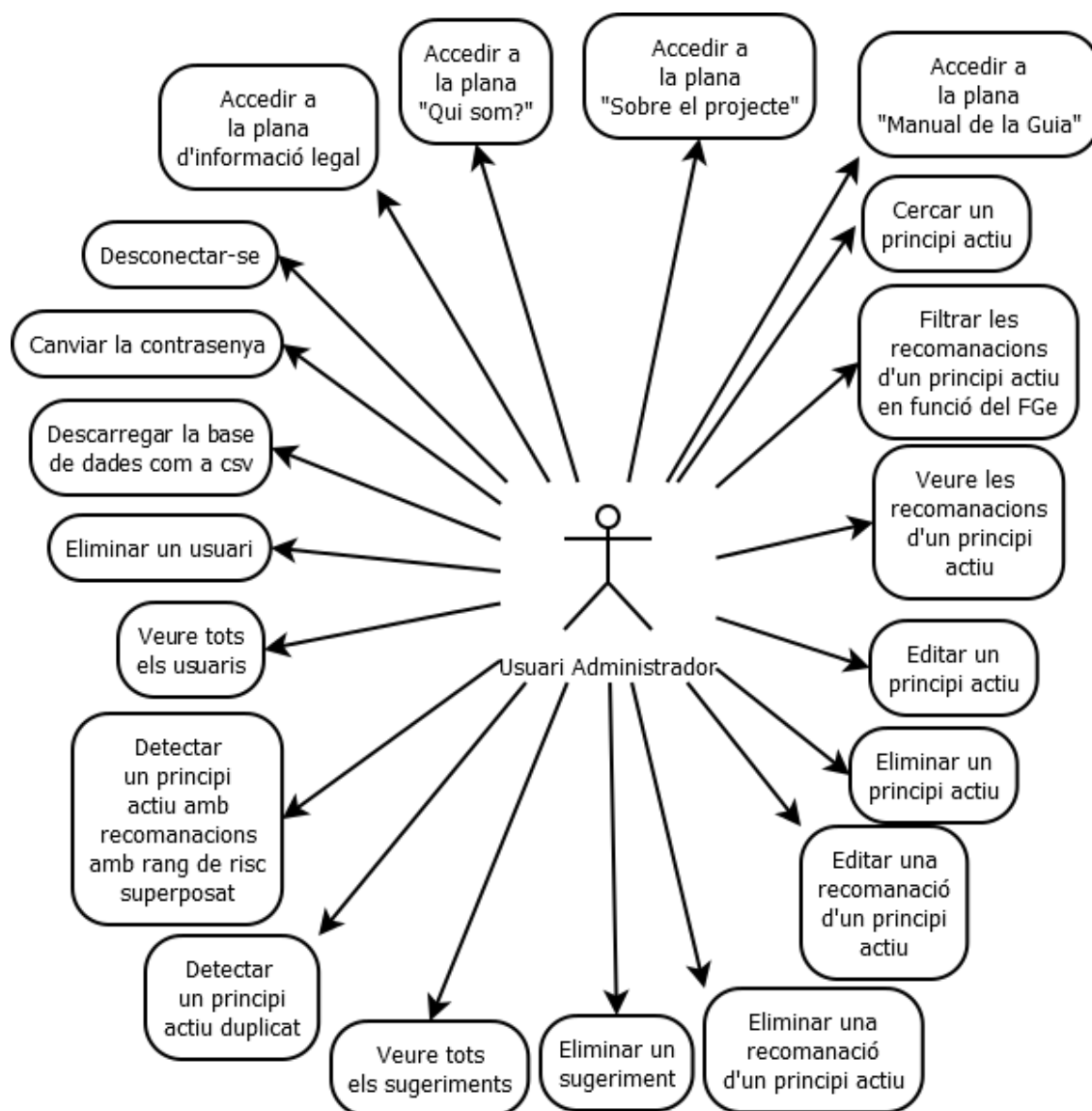


Figura 3: Casos d'ús de l'usuari administrador

L'usuari administrador és l'únic amb permisos per alterar els principis actius guardats a la base de dades.

4.4 Requeriments no funcionals

Seguretat

La seguretat de la base de dades recau en darrera instància en l'administrador del servidor on estigui allotjat la mateixa. Es podria encriptar totes les dades de la base de dades amb un algorisme bidireccional però això implicaria haver de desencriptar les dades a cada accés la qual cosa impactaria severament en el rendiment. Per aquest motiu l'única informació encriptada serà les contrasenyes dels usuaris amb

un Hash unidireccional per a fer impossible la recuperació d'aquesta informació.

Pel que fa a la seguretat a l'autenticació s'ha implementat un sistema d'autenticació mitjançant JSON Web Tokens, JWT. Els JSON Web Tokens és un mètode industrial estàndard, RFC 7519; i obert per assegurar permisos entre dues entitats [9]. Els JWT es poden signar mitjançant una clau secreta, algorisme HMAC; o mitjançant un parell de claus, una pública i una altra privada, amb l'algorisme RSA. Aquests tokens poden estar encriptats a més de signats. La seguretat d'un JWT recau en la signatura. Al ser signat amb una clau secreta únicament disponible a l'entitat que l'ha creat es pot verificar la legibilitat del token comprovant la signatura. És un mètode d'autenticació bastant utilitzat ja que l'usuari tan sols necessita enviar la contrasenya un sol cop, durant l'autenticació i després simplement ha d'enviar el JWT al servidor per determinar la seva identitat. Una altra avantatge dels JWT es la seva mida reduïda el qual fa possible enviar-los a dins la capçalera de la crida HTTP.

Rendiment i escalabilitat

Tant Angular com Lumen son inherentment més ràpids i eficients que AngularJs i Laravel. Per tant si abans l'aplicació ja era ràpida ara ho serà encara més. El punt més ineficient de la pàgina, la pàgina de visualització d'usuaris degut al fet que realitzava la paginació a la part de client s'ha solucionat movent la paginació a la part de servidor.

Pel que fa al tema d'escalabilitat tot depèn del servidor on estigui muntat. Des de la part de l'aplicació l'únic que es pot intentar es assegurar que la web i el *backend* executin les seves tasques el més eficientment possible per a consumir la menor quantitat de recursos. La responsabilitat de controlar el volum de peticions recau al servidor on estigui allotjada la web.

Accessibilitat

Degut al canvi d'AngularJs a Angular juntament amb la migració de Bootstrap 3 a 4 s'ha hagut de refer tota la part de client. S'ha aprofitat aquest començament de zero per a dissenyar-la pensant en que es visualitzaria en pantalles petites aplicant el patró Responsive Web Design, RWD. Aquest patró de disseny consisteix en adaptar l'aparença de les planes web al dispositiu que s'estigui utilitzant per visitar-les. Això es necessari ja que avui en dia s'utilitzen multitud de dispositius diferents per navegar per internet: mòbils, tablets, ordinadors, llibres electrònics, portàtils. Aquesta tecnologia pretén que, amb un únic disseny, tot es vegi correctament a qualsevol dispositiu. [10]

S’ha aconseguit que totes les pàgines es veiessin correctament en qualsevol mida de pantalla. Tot i això, degut a que en els formularis de principis actius s’ha d’escriure bastant l’experiència d’usuari serà millor en ordinadors o dispositius que disposin de teclat. A la resta de pàgines en canvi l’experiència d’usuari serà igual de bona independentment de la mida de la pantalla on es vegi.

Facilitat d’ús

És imprescindible mantenir la facilitat d’ús de la web i més tenint en compte que serà utilitzada en dispositius mòbils on la reduïda mida de la pantalla provoca que l’usuari no pugui veure tota la pàgina de cop. L’objectiu principal és que a cada interacció es disposi de tota la informació necessària en pantalla per a poder dur-la a terme. Per a ajudar a que els usuaris es familiaritzessin amb les noves funcions s’ha pujat una versió *demo* de la web a un servidor públic, Heroku [11], unes setmanes abans de fer la pujada definitiva al servidor d’hostatgeria web de la UB.

Un altre objectiu relacionat amb la facilitat d’ús, no dels usuaris si no dels futurs desenvolupadors de la web, es la creació de la guia per llençar un servidor capaç de córrer la plana web en una maquina virtual i també per pujar-ho a un servidor a Heroku, donat que es una plataforma que ofereix un servidor públic gratuït bastant senzillet però prou potent per a *demos*.

5 Planificació

5.1 Planificació inicial

A continuació es mostra el Diagrama de Gantt amb la planificació inicial d'aquest projecte.

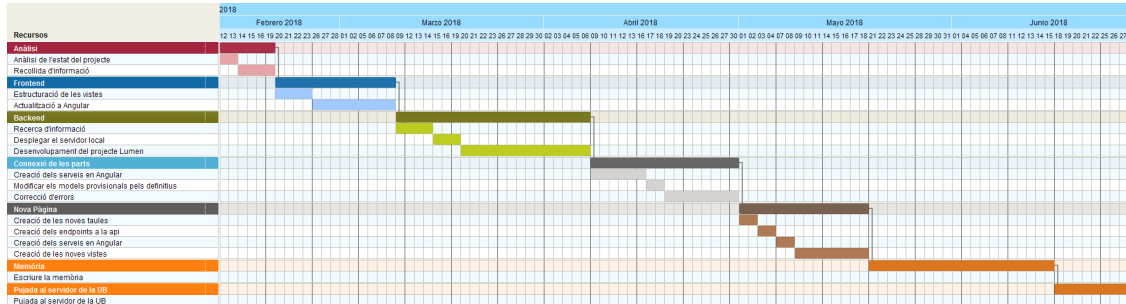


Figura 4: Diagrama de Gantt

La idea inicial ha estat dedicar una setmana a analitzar l'estat del projecte i a recopilar informació per a assegurar que les tecnologies escollides a l'inici del treball fossin les adequades per a dur-lo a terme. Tot seguit s'esperava dedicar unes dues a tres setmanes per a actualitzar la part de client, de AngularJS a Angular 5 i després dedicar el més següent a desenvolupar la part de servidor en Lumen, primer recopilant informació per a saber com utilitzar-lo i com integrar-lo amb les tecnologies d'autenticació decidides i també veure quines parts del codi antic es podien reaprofitar ja que Lumen estava basat en Laravel. Tot seguit es tenia pensat dedicar tres setmanes a integrar les dues parts, modificant les parts necessàries del codi Angular per a utilitzar la Api desenvolupada en Lumen. Finalment, la darrera tasca de desenvolupament era la creació de la nova pàgina per a guardar les dades dels pacients i complir així el principal objectiu d'aquest projecte.

Una volta acabat aquest projecte es tenia pensat disposar d'un més per a escriure la memòria i uns pocs dies per a realitzar la pujada al servidor d'hostatgeria web de la UB.

5.2 Compliment de la planificació

Al acabar aquest projecte s'ha vist que s'ha subestimat la dificultat de l'actualització a Angular 6. Tot i que es cert que el desenvolupament en Angular és bastant senzill si es té ja una mica d'experiència el principal problema es que no s'ha pogut reaprofitar pràcticament res del codi que ja estava desenvolupat. Això ha provocat que en comptes de dues a tres setmanes la fase de client s'allargués fins a casi dos mesos, triplicant l'estimació de temps inicial. La part de servidor, en canvi, si que es s'ha ajustat al temps previst. El problema sorgeix a l'hora de fer la integració, que un cop més s'ha subestimat la dificultat i s'ha acabat allargant una setmana més de l previsió inicial. Tot i això, veure i solucionar tots els problemes que han

sorgit ha permès realitzar la nova pàgina en un temps record tardant únicament una setmana. La resta del temps ha estat dedicat a realitzar la memòria i s'ha aplaçat la pujada fins al Juliol.

6 Disseny

6.1 Definició del model

A continuació es mostra el model de dades que s'utilitzen a les noves funcionalitats implementades, és a dir, els registres de consultes i la pàgina de seguiment de pacients.

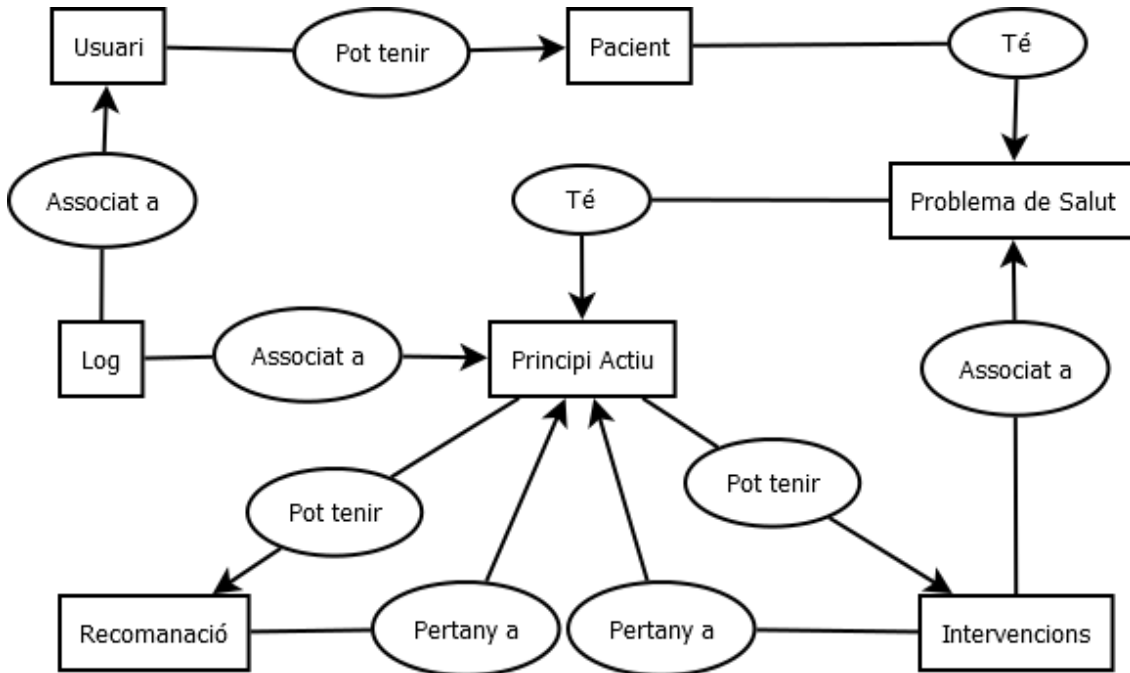


Figura 5: Definició del model treballat

Primer definim els atributs del registre. Un registre estarà associat a un usuari concret i registrarà la consulta d'un principi actiu en un moment determinat. Si l'usuari ha indicat un filtratge glomerular estimat per a filtrar les recomanacions del principi actiu també el guardarem.

Tot seguit es defineixen els atributs d'un pacient. Un pacient rep seguiment d'un usuari registrat. Aquest usuari necessita guardar l'edat, el sexe, l'índex de massa corporal, IMC; el filtratge glomerular estimat, FGe; les al·lèrgies, en el cas que en presentés alguna, i la zona territorial del pacient. A més s'han de guardar els problemes de salut del pacient així com els principis actius, els seus efectes sobre la seva salut i les intervencions, en el cas que fossin necessàries, dels medicaments que estigui prenent per a tractar dit problema de salut.

Un problema de salut d'un pacient està definit per la data en que va començar la dolència, la descripció dels símptomes d'aquesta malaltia i si preocupa o no al pacient.

Un tractament, és a dir, un principi actiu que està prenent un pacient per tractar un dels seus problemes de salut estarà definit per un principi actiu, si es necessari, eficient o segur; el nivell de risc que presenta per al pacient així com una descripció del problema relacionat amb el medicament, PRM, si escaigués. A més, es vol guardar el conjunt d'intervencions que son necessàries fer per assegurar l'òptima ingesta d'aquest principi actiu per a aquest pacient en concret.

6.2 Model Entitat-Relació

En aquest apartat es defineixen les relacions entre els diferents models de l'aplicació que afecten a les noves funcionalitats implementades.

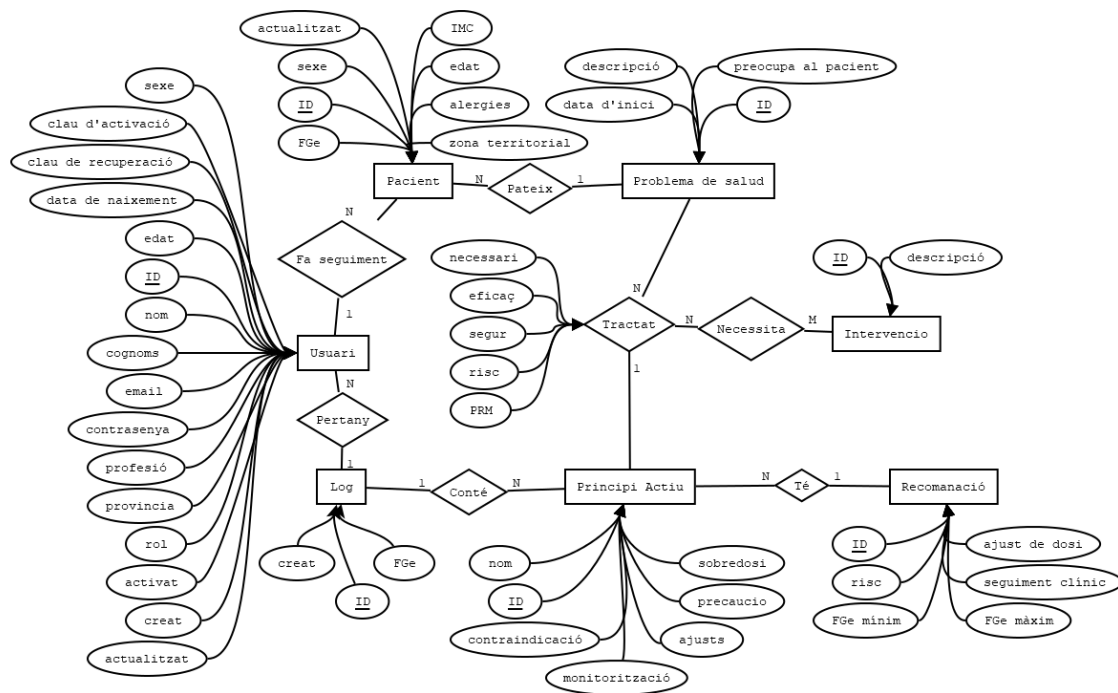


Figura 6: Model Entitat-Relació segons la notació de Chen

Començarem per les relacions que afecten als registres de cerca. Un registre de cerca té un com a mínim un usuari i, també com abans, com a mínim un principi actiu. Tant els usuaris com els principis actius poden tenir cap, un o més registres de cerca. És a dir, entre el registre i els usuaris tenim una relació N a 1 i ídem per als principis actius.

Seguim amb les relacions que afecten als pacients. Un usuari registrat pot fer un seguiment a cap, un o més pacients. Tenim per tant una relació 0 a N entre els usuaris i els pacients. Cada pacient tindrà un o més problemes de salut, és a

dir una relació 1 a N entre pacients i problemes de salut. Cada problema de salut estarà sent tractat per mitjà d'un o més principis actius, per tant tenim una relació 1 a N un altre cop. Cada un d'aquests principis actius tindran un conjunt de intervencions per assegurar la òptima ingesta del mateix. Per tant estem davant d'una relació M a N.

6.3 Definició del model relacional

Finalment es mostra el model relacional de la base de dades MySQL utilitzat a l'aplicació.

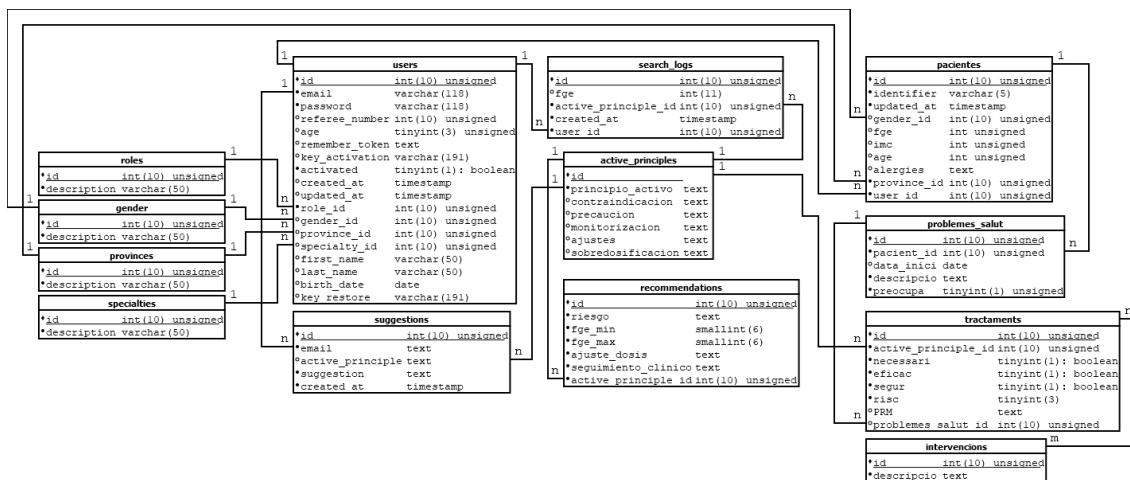


Figura 7: Model Relacional de la base de dades

Abans de res es comentaran els canvis que s'han fet a les taules existents durant aquesta etapa de desenvolupament.

A la taula de principis actius s'han eliminat les columnes *monitoritzacin_no* i *ajustes_no*. A més les columnes *monitorizacion_si* i *ajustes_si* s'han renomnat *monitorizacion* i *ajustes*. Aquesta decisió s'ha pres per unificar les dades que es mostraven a la part de client i les que es guardaven ja que a la part de client tant *monitorizacion_no* com *monitorizacion_si* es mostraven juntes en un mateix camp i al formulari d'edició tan sols s'introduïa *monitorizacion_si*, ídem per al camps *ajustes*. És més, després d'analitzar la base de dades s'ha vist que el camp *monitorizacion_no* sempre estava a nul. Abans d'eliminar les columnes es van concatenar els seus valors per evitar la pèrdua d'informació. L'únic problema es que es un canvi irreversible ja que no es podrà saber quins camps estaven junts i quins no. Finalment, es va considerar eliminar les files que tinguessin principis actius duplicats, es a dir que tinguessin el nom repetit. Després d'analitzar les files conflictives es va veure que no estaven sempre totalment duplicades i per evitar una possible pèrdua d'informació es va decidir implementar la funcionalitat de solucionar files duplicades a la part de client i millorar les validacions de les entrades a la part de

servidor per evitar la reintroducció de principis actius duplicats.

A la taula de recomanacions també s'han fet canvis. El primer i més evident ha estat eliminar la columna *principio_activo* ja que es podia inferir a partir de la clau forana de principis actius. Després s'han modificat les columnes *fge_min* i *fge_max* perquè els valors menors a 0 i superiors a 140 tinguessin 0 i 140 respectivament. És va considerar modificar aquesta taula per eliminar recomanacions amb rangs de Fge superposats però després d'un breu anàlisis es va veure que es podria perdre informació i per tant es va implementar com a funcionalitat a la banda de client. També es van implementar validacions tant a la part de client com a la part de servidor per evitar que es tornessin introduir dades amb rangs superposats.

Seguim amb la taula de suggeriments. En aquesta taula simplement s'ha introduït la columna de data de creació. En el moment de fer la migració per afegir aquesta taula es fica com a valor per defecte la data de la migració ja que serà una columna no nul·la.

A continuació s'explicarà la nova taula de registres de cerca. Aquesta taula té el propòsit de monitorar les consultes que fan els usuaris a la guia per tal de poder fer un anàlisi de dades i estudiar els principis actius que s'estiguin cercant més freqüentment. Per cada consulta guardam l'identificador de l'usuari que l'ha fet, l'identificador del principi actiu que ha cercat, la data en que s'ha fet la cerca i finalment el nivell de filtratge glomerular estimat, si l'usuari l'ha indicat per filtrar les recomanacions del principi actiu,

Passam a les noves taules introduïdes per a la nova pàgina de seguiment de pacients. En primer lloc, la pròpia taula de pacients. En aquesta es guarda en primer lloc l'identificador de l'usuari que l'ha introduït. També guardam un identificador de 5 lletres perquè l'usuari tingui una forma d'identificar-lo després d'introduir les dades. La resta de dades són el sexe, guardat com una clau forana que apunta a la taula de sexes, el nivell de filtratge glomerular estimat del pacient, el seu índex de massa corporal, l'edat, les seves al·lèrgies i la zona territorial on viu.

Seguim amb la taula de problemes de salut. Primer de tot guardam l'identificador del pacient per a tenir la relació amb la taula de pacients. També es guardarà la data en que va sorgir aquest problema de salut, una petita descripció del mateix i un booleà indicant si preocupa al pacient.

La següent taula és la de tractaments. Aquesta taula relaciona un principi actiu amb un problema de salut a la vegada que guarda dades específiques d'aquest tractament. Aquestes són si es necessari, eficient i/o segur per al pacient i una breu descripció si escau del problema relacionat amb el medicament. Cada tractament pot tenir un conjunt de intervencions que es guardaran en una taula apart. Per tal de relacionar les dues, ja que es tracta d'una relació M a N es crearà una taula

intermèdia que contindrà l'identificador del tractament i l'identificador de la intervenció.

La darrera taula, la de intervencions, es bastant senzilla, consisteix en un identificador i una descripció de la intervenció.

7 Tecnologies utilitzades

7.1 Servidor Web

Aquest punt ha estat un dels principals problemes de tot el projecte degut a les característiques del servei d'hostatgeria de la UB que obliguen a treballar amb una versió molt concreta de PHP i sense la possibilitat de baixar-se les dependències directament al servidor al no tenir Composer. La guia es troba allotjada al servidor públic de la Universitat de Barcelona amb el servei d'hostatgeria gratuït que aquesta ofereix per a petits portals web que requereixin un consum de recursos raonable [8]. Aquest servei recentment ofereix la versió 7.0.27 de PHP, però la web es troba actualment allotjada en un servidor amb la versió 5.3.27 que es va llançar l'11 de Juliol de 2013 i que va deixar de rebre suport el 14 d'Agost de 2014 [12, 13]. La versió que suporta actualment el servidor de la UB es va llançar el 4 de Gener de 2018 i no encara no està previst que deixi de rebre suport. El sistema operatiu del servidor es un Linux amb *kernel* 3.0, una base de dades MySQL versió 5.5.19 i un servidor Apache2 versió 2.2. De tots aquests requisits el més important es la versió de PHP.

Per a l'entorn de desenvolupament local s'ha treballat amb la versió 7.2.2 de XAMPP per Windows, és a dir, PHP versió 7.2.2, Apache2 versió 2.4.29 i MariaDB versió 10.1.30. Ja de primeres es pot veure que les versions no són les mateixes però no es un problema, perquè com ja he dit, a excepció de la versió de PHP la resta no ens afecta pel que respecta al desenvolupament.

Primer de tot i el més important, no tenim la mateixa base de dades. Però no ens hem de preocupar per incompatibilitats ja que MariaDB es un *fork* de MySQL desenvolupada pels propis creadors de MySQL i que ha sorgit arrel de la compra de MySQL per part de Oracle [14, 15]. Tot i que és cert que hi podria haver certs aspectes on puguin ser incompatibles no han afectat al desenvolupament del projecte. De fet Eloquent, l'ORM de Lumen detecta MariaDB amb el *driver* de MySQL.

Seguim amb Apache2, encara que estem dues *minor version* per damunt del de la UB l'únic problema que ha sorgit ha estat amb el *driver sendmail* de la funcionalitat d'enviar correus degut a que és necessiten configuracions addicionals per fer-lo servir a Windows. Aquest problema s'ha solucionat utilitzant el *driver smtp*.

Finalment, la versió de PHP, aquí es on més problemes poden sortir i que durant el desenvolupament no s'ha tingut amb compte al fer el projecte de zero. S'ha desenvolupat amb la versió 5.6 de Lumen, és a dir, la darrera. El problema es que aquesta versió només es compatible amb la versió 7.2 de PHP. Per sort la versió immediatament inferior de Lumen, la 5.5 és compatible amb la versió de PHP del servidor de la UB [16] i no s'ha utilitzat cap funció específica de la darrera versió de Lumen. De fet quan es busca documentació de Lumen s'ha de revisar quina versió

s'està consultant ja que no sempre et mostra la documentació de la més recent.

Una volta acabat es va muntar un servidor en una maquina virtual Linux amb la darrera versió LTS d'Ubuntu, Ubuntu Server versió 18.04 LTS [17]. Aquí es va fer una primera prova per veure si el que s'havia desenvolupat funcionaria en el servidor de la UB però no es van tenir en compte les versions ni els requeriments que tenia Lumen per a funcionar a Apache. Això va provocar bastant errors i al final que el projecte no funcionés. En aquest moment es va prendre la decisió de no perdre més el temps amb aquest servidor i pujar el projecte a un servidor públic per dos motius. El primer, que el projecte no funcionava segurament degut a un problema de configuració del propi Apache i, el segon, que un servidor públic permetria que el personal del Grup pogués provar la web abans de pujar-la al servidor definitiu. Es va escollir utilitzar el servidor públic de Heroku. Heroku es una plataforma al núvol basada en un sistema de maneig de contenidors, amb serveis de dades integrats i un potent ecosistema, per a desplegar i córrer aplicacions modernes. L'experiència de desenvolupador de Heroku es una aproximació centrada en aplicacions de la distribució de programari, integrat amb les eines de desenvolupament i els fluxos de treball més popular avui en dia [18]. Apart de la facilitat de desplegament que ens dona Heroku, literalment es pujar el projecte a una branca de GitHub, també ens permet escollir la versió que volen de PHP ja que disposen de les següents versions de PHP 5.6, 7.0, 7.1 i 7.2, això ens permet provar el projecte amb la versió més similar possible a la que disposa actualment el servidor de la UB.

Finalment, la darrera barrera a l'hora de desplegar la web al servidor de la UB es que no té instal·lat Composer. Composer es una eina per a manejar dependències en un projecte de PHP. Permet declarar les llibreries necessàries per al projecte i s'encarrega de instal·lar-les i/o actualitzar-les [19]. Això vol dir que abans de poder pujar el projecte es necessari tenir un entorn amb PHP 7.0.27 instal·lat per a poder baixar les dependències correctes. Notar que només és necessita la versió de PHP ja que podem provar el funcionament de l'aplicació a Heroku. Aquest entorn es simplement per baixar les dependències abans de fer la pujada al servidor de la UB.

7.2 Lumen

Lumen es el *framework* de PHP que ha substituït a Laravel en aquest projecte. Abans d'explicar els motius explicarem breument què es Lumen. Lumen es la solució de Laravel pensada per a implementar Apis i micro-serveis. És, per tant, un *micro-framework* basat en Laravel però més ràpid al prescindir de moltes característiques que Laravel ja porta integrades i que no son necessàries per a aquests tipus de projectes. Lumen va néixer durant la creació de Envoyer, una eina per desplegar aplicacions PHP, per l'autor de Laravel, Taylor Otwell, degut a que va haver d'implementar dos micro-serveis que necessitava que fossin el més ràpid possibles i va trobar a faltar la facilitat que oferia Laravel per la gestió d'errors i connexió a base de dades que no oferien els competidors, Slim i Silex [20].

Ja és pot veure el motiu pel qual s'ha escollit Lumen en comptes de Laravel a l'hora de reconstruir la REST API de l'aplicació. El major rendiment i menor consum de recursos de Lumen juntament amb la mateixa extensibilitat de Laravel i la quasi total compatibilitat amb el codi anterior han estat els factor decisius per a fer el canvi.

7.3 Composer

Composer, com s'ha explicat anteriorment, es un gestor de dependències per a PHP. Es una eina imprescindible per al desenvolupament de Lumen ja que no només s'en-carrega de crear l'esquelet de l'aplicació Lumen si no de baixar tots els paquets que necessita Lumen per a funcionar mentre s'assegura la compatibilitat entre les ver-sions instal·lades. Les dependències del projecte es declaren a un fitxer JSON, `composer.json`, indicant la versió aproximada que es vol utilitzar. No es recomana mai indicar versions concretes a no ser que es tingui un motiu de pes per fer-ho, com per exemple indicar la versió concreta de PHP que utilitza el servidor de la UB. Les versions exactes que es baixen es guarden al fitxer `composer.lock` juntament amb tota la informació necessària per a partir d'aquest fitxer poder baixar totes les dependències del projecte. Heroku utilitza aquest fitxer per descarregar i instal·lar al servidor les dependències que utilitza el projecte.

7.4 Bootstrap

Bootstrap es un conjunt d'eines open source per a desenvolupat amb HTML, CSS i Javascript [21]. En aquest projecte s'han utilitzat únicament les classes CSS que ofereix ja que els components que requereixen de Javascript no son compatibles amb Angular 2. Per això s'han utilitzat els components compatibles realitzats per els desenvolupadors de la llibreria de components de Bootstrap per a AngularJs, part d'aquest equip de desenvolupadors pertanyen a Angular; Ngb-Bootstrap [22].

7.5 Angular

Angular es una plataforma que facilita el desenvolupament d'aplicacions web. Com-bina plantilles declaratives, injecció de dependències, eines de testeig e2e i integra les *millors pràctiques* per a solucionar reptes del desenvolupament [23]. Angular fa ús de NodeJs per a fer funcionar la seva interfase de línia de comandes, Angular CLI. Aquesta permet entre altres coses desplegar el servidor de desenvolupament alhora que monitorea els canvis fets i els aplica en temps real, muntar i compilar l'aplicació així com crear l'esquelet de l'aplicació.

Anem a veure les diferències entre AngularJs i Angular 2+. La principal és la filosofia que segueix cadascú, mentre AngularJS segueix la tradicional estructura MVC, on el controlador gestiona com interactua l'usuari amb la pàgina i actua de pont entre la vista i el model, la vista representa la interfície gràfica i el model es una representació abstracta de les dades.

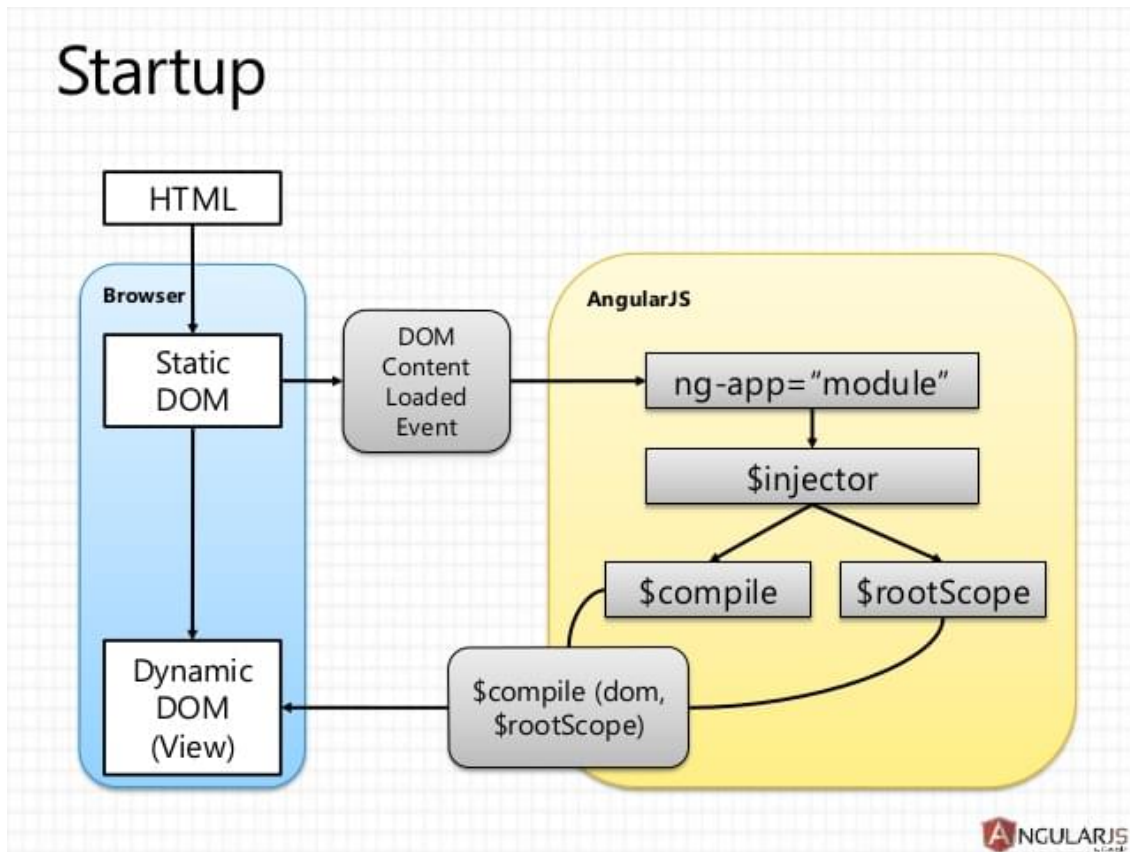


Figura 8: Estructura d'una aplicació AngularJS.

Font: <https://www.sitepoint.com/angularjs-vs-angular/>

Angular 2+ en canvi abraça la filosofia de estructura basada en components. Cada aplicació Angular té almenys un component conegut com component arrel. Cada component té una classe associada responsable de gestionar la lògica de negoci i una plantilla que actua com la capa de vista. Múltiples components poden ser ajuntats per a crear un modul i cada modul conforma una unitat funcional per ell sol.

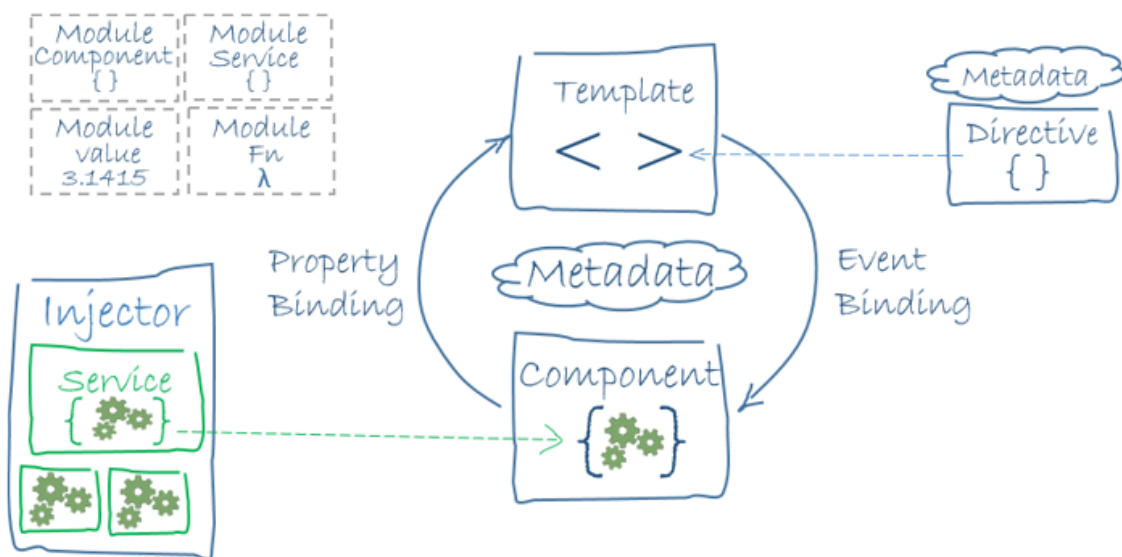


Figura 9: Estructura d'una aplicació Angular 2+.

Font: <https://www.sitepoint.com/angularjs-vs-angular/>

Una altra de les diferències entre AngularJs i Angular 2 es el llenguatge utilitzat. Mentre que AngularJs només suporta Javascript, això és un punt a favor del framework ja que per utilitzar-lo únicament es necessita tenir coneixements bàsics de Javascript el que fa que la seva corba d'entrada sigui molt més suau. Angular 2 en canvi va donar el salt a Typescript, un superset sintàctic de Javascript que compila a simple Javascript. Un dels principals avantatges de Typescript davant Javascript es que permet implementar validació de tipus estàtica opcional. Aquest fet va ser el que va fer decantar a l'equip d'Angular d'escollir-lo com a llenguatge per defecte ja que fa el codi Javascript més predictable.

Finalment, i un dels principals motius que van fer escollir Angular 2 en comptes d'AngularJs es que la majoria de problemes que presenta el segon estan relacionats al rendiment.

7.6 NodeJs i npm

NodeJs es un entorn d'execució per a Javascript muntat amb el motor de Javascript v8 de Chrome. Utilitza un model d'operacions E/S no bloquejant i orientat a esdeveniments, que el fa lleuger i eficient. El seu ecosistema de paquets, npm, es l'ecosistema més gran de llibreries de codi obert del món [24].

Com ja s'ha dit anteriorment, Angular utilitza Node per a córrer la seva CLI i també npm com a gestor de dependències. De forma similar a com ho fa Composer, npm utilitza un l'arxiu package.json per entre altres coses definir les dependències necessàries i el fitxer package-lock.json per a guardar totes les llibreries amb les versions exactes que ha instal·lat.

8 Desenvolupament

Durant el desenvolupament es tenien dues carpetes separades, una amb tot el codi Angular i l'altre amb tot el codi Lumen. Mentre que l'aplicació Angular podia estar a on volgués ja que corria en un servidor Node proporcionat pel propi Angular CLI, l'aplicació de Lumen havia d'estar a la carpeta htdocs de XAMPP.

Dit això passam a explicar l'estructura de l'aplicació Angular, ja que es la primera que es va realitzar, es seguirà explicant l'estructura de l'aplicació Lumen, realitzada una volta es va tenir una primera versió del frontend. Finalment, s'explicarà el procés d'integració per a fer funcionar la pàgina web en un sol directori del servidor.

8.1 Aplicació Angular

Estructura

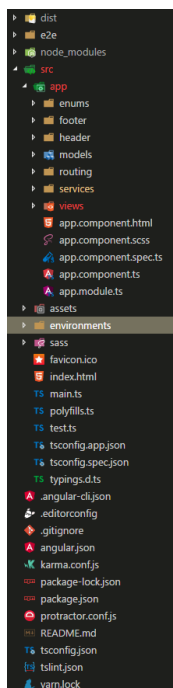


Figura 10: Estructura de l'aplicació Angular

A la figura es pot veure l'estructura de carpetes de l'aplicació Angular acabada. De totes les carpetes que hi ha les més importants son `dist` i `app`. `Dist` es la carpeta que es crea al efectuar la comanda `build` d'Angular CLI i conté l'aplicació Angular compilada. La carpeta `src` conté tot el codi font mentre que les carpetes `e2e` i `node_modules` contenen els casos de test *end to end* i les llibreries necessàries, respectivament.

Dins `src` trobam la carpeta `app` que conté tots els components, models i serveis que s'han creat per a l'aplicació. També hi trobam els recursos estàtics, dins `assets`, els fitxers de variables d'entorn, dins `environments` i els arxius SASS que contenen els estils globals de la pàgina, dins la carpeta `sass`. Altres arxius importants que es troben dins aquesta carpeta son el `package.json` i `package-lock.json`, que ja s'ha explicat per a que serveixen a l'apartat de NodeJs, l'`index.html`, que és el punt d'entrada de l'aplicació Angular.

Finalment arribam a `App`. Aquí tenim per ordre els `enums`, és a dir, llistes estàtiques de constants, que en aquesta aplicació només conté els rols. Segueixen els components de `footer` i `header` que no s'han inclòs dins la carpeta de `views` ja que son elements del template global. Arribam a la carpeta `models` que conté les diferents classes contenidores de dades que s'han creat per a l'aplicació. Aquestes classes s'encarreguen d'encapsular les dades que ens arriben en forma de JSON des de la API REST. Tot seguit trobam

el mòdul de routing que conté totes les rutes de l'aplicació Angular, no conté les rutes de la API REST ja que aquestes es troben definides a l'aplicació Lumen, a la darrera secció d'aquest capítol s'explica com es gestiona perquè cada ruta utilitzi el sistema de routing que li pertoca. A continuació trobam el mòdul de serveis que conté totes les classes encarregades de fer peticions a la API REST. Per acabar tenim el mòdul de views que conté els components que Angular s'encarregarà de injectar o treure dinàmicament segons la ruta en que ens trobem.

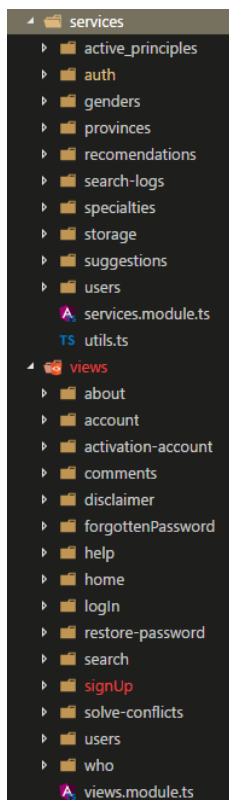


Figura 11: Continguts de *views* i *services*

A la figura es pot veure amb més detall els serveis i les vistes implementades, No vol dir que només s'hagin implementat aquest components ja que alguna de les vistes més complicades com *search* conté diversos submòduls. No es mostren desplegats perquè aleshores la imatge s'allargaria durant varies pàgines.

A continuació s'explicaran les diferents classes utilitzades per a fer el projecte. Es seguirà el mateix ordre de les carpetes i així es podrà explicar per damunt les característiques de Typescript.

Començam amb l'enum, que es defineix com es pot veure a continuació. Ja es pot veure una de les característiques de Typescript, la paraula clau *export* que indica que la enum definida pot ser exportada en un altre arxiu. *Export* també és una paraula clau de Javascript definida a l'especificació ECMAScript 2015 6th Edition, ECMA-262 [25].

```
export enum <EnumName> {
    <EnumConst> = <EnumValue>,
    ...
}
```

Seguim amb els components, un component a Angular es una classe Typescript anotada de la següent manera:

```
@Component({
  selector: '<selector-name>',
  templateUrl | template: <url | template>,
  styleUrls | styles: <url-array | styles>
})
export class <componentName> {
  ...
}
```

Amb aquesta anotació el compilador d'Angular sap que necessita importar tot un conjunt de codi per tal de fer tres coses. Per una part enllaçar el codi Javascript

compilat amb el codi html del template indicat, aquest template pot estar definit inline dins el propi arxiu Typescript (.ts) o en un arxiu a part indicant la URL. Per l'altra ha de enllaçar els estils indicats al codi HTML enllaçat, cal notar que els estils indicats al fitxer d'estils o definits inline només apliquen al component on s'ha enllaçat.

The styles specified in @Component metadata apply only within the template of that component. [26]

Finalment, amb el selector se li indica el nom que tindrà l'element en el DOM, Document Object Model, el qual ha de ser substituït per el template indicat.

Abans de seguir amb els components amb més profunditat s'explicaran els serveis. A continuació es pot veure com es defineix un servei amb Typescript:

```
@Injectable()
export class <serviceName> {
  ...
}
```

Ja podem veure una altra de les anotacions d'Angular, Injectable. Aquesta anotació serveix per indicar a Typescript que aquesta classe pot ser proporcionada mitjançant un injector de dependències, en el nostre cas el d'Angular. És similar als Beans de Spring en Java. Per no allargar-nos massa, un injector de dependències és una classe contenidora de instàncies d'altres classes injectables. D'aquesta forma es té un lloc on s'instancien un sol cop aquestes classes i la resta de classes que les necessiten les poden agafar d'aquest injector. Tot aquest procés es executa en segon pla per Angular. A més Angular té la particularitat de que permet tenir més d'un injector de dependències, ja que aquests es defineixen en els mòduls. A continuació es mostra la nomenclatura de l'anotació NgModule.

```
@NgModule({
  imports: ...,
  declarations: ...,
  providers: ...,
  exports: ...
})
export class <moduleName> { }
```

Aquests no són els únics atributs que té el mòdul d'Angular però són els que ens interessa per acabar d'explicar els serveis i després fé una petita explicació del propòsit dels mòduls. En primer lloc, l'atribut providers defineix els singletons que contindrà l'injector de dependències d'aquest mòdul. Si altres mòduls importen un mòdul que proveeixi serveis, els components declarats en el primer tindran accés a l'injector de dependències del segon. Passam a definir la resta d'atributs, començant per el de declarations. A aquest atribut s'indiquen els components que conté el mòdul,

i, per tant, el selector del component només funcionarà dins altres components del mateix mòdul. L'atribut `exports`, en canvi, declara els components que el mòdul fa accessibles a altres mòduls que l'importin i que per tant podran ser utilitzats en components d'altres mòduls. I per acabar, l'atribut `import` indica els mòduls que han estat importats dins un mòdul.

Després d'aquest no tan breu incís seguim amb el component. En concret ens centrarem en com s'injecten serveis a una classe. Per indicar que una classe necessita un servei proveït per un mòdul s'ha de declarar el servei com a paràmetre del constructor de la classe, d'aquesta forma Angular automàticament injectarà el servei al component quan sigui creat.

Tot seguit, es tractarà l'enllaç d'atributs del component al template. Angular té dos tipus d'enllaç, unidireccional i bidireccional. Un enllaç unidireccional vol dir que les dades només viatgen en un sentit, d'un component pare a un component fill o a la inversa. Això implica que els canvis a les dades tan sols seran detectats i actualitzats en el sentit en que viatgin. Per exemple, si una dada passa del pare al fill, si el pare canvia la dada el fill veurà el canvi, però en canvi, si el fill modifica la dada el pare no ho detectarà. Una altra característica es que, mentre que els canvis cap avall, és a dir, de pares a fill, es propaguen automàticament, no es així quan es en l'altre sentit, en aquest cas s'han d'enviar manualment cap amunt. Un enllaç bidireccional, aleshores, no és més que dos canals unidireccionals enllaçats a un únic atribut. Per indicar un enllaç unidireccional descendent s'indica entre `[]` mentre que per indicar un enllaç unidireccional ascendent s'indica entre `()`. Un enllaç bidireccional s'indica combinant les dues formes anteriors, és a dir, `[(())]`.

Per acabar, faltaria explicar les directives. Per indicar una Directiva s'indica amb l'anotació `Directive` d'Angular. Una directiva es una classe que interactua amb el DOM, ja en coneixem una, els components, és a dir, directives amb templates associades. Però n'hi ha d'altres tipus, directives d'atribut i directives estructurals. Una directiva d'atribut és una classe que modifica l'aparença o comportament d'un atribut, component o una altra directiva. En canvi una directiva estructural és una directiva que canvia l'estructura del DOM afegint o eliminant elements. Un exemple de directiva d'atribut que es la directiva `NgClass` que permet afegir o eliminar classes CSS dels elements del DOM en funció d'altres atributs. Aquesta es una directiva que ofereix Angular de forma nativa. De les directives estructural les més utilitzades son `*ngFor` i `*ngIf`. La primera permet repetir el template que conté un nombre determinat mentre que la segona permet mostrar o ocultar el template associat en funció d'un atribut.

Vistes

Pel que fa a la part gràfica, s'ha intentat mantenir l'aspecte que tenia l'aplicació canviant únicament les parts que perjudicaven al responsive de la pàgina o que havien deixat de tenir sentit arrel de les noves funcionalitats.

Començarem pel template bàsic compost per el header i el footer. El header està implementat mitjançant les classes CSS navbar de Bootstrap i controla la navegació entre les diferents pestanyes de la pàgina. Els enllaços es gestionen per mitjà de navlinks que connecten els links amb el router d'Angular. Quan un usuari no està registrat té accés a la pàgina principal, on es veu el títol de la pàgina i també a les pàgines d'informació sobre el projecte i sobre els participants i col·laboradors en el mateix. També té disponibles les pàgines de registre i d'autenticació. Aquestes dues pàgines són les que s'han canviat ja que abans eren modals. A més, quan la pantalla es prou petita la navbar es converteix en un menú desplegable. Els canvis més evidents són que el botó es veu correctament quan la pantalla es fa petita i que al clicar a un enllaç o fora del desplegable aquest s'amaga.

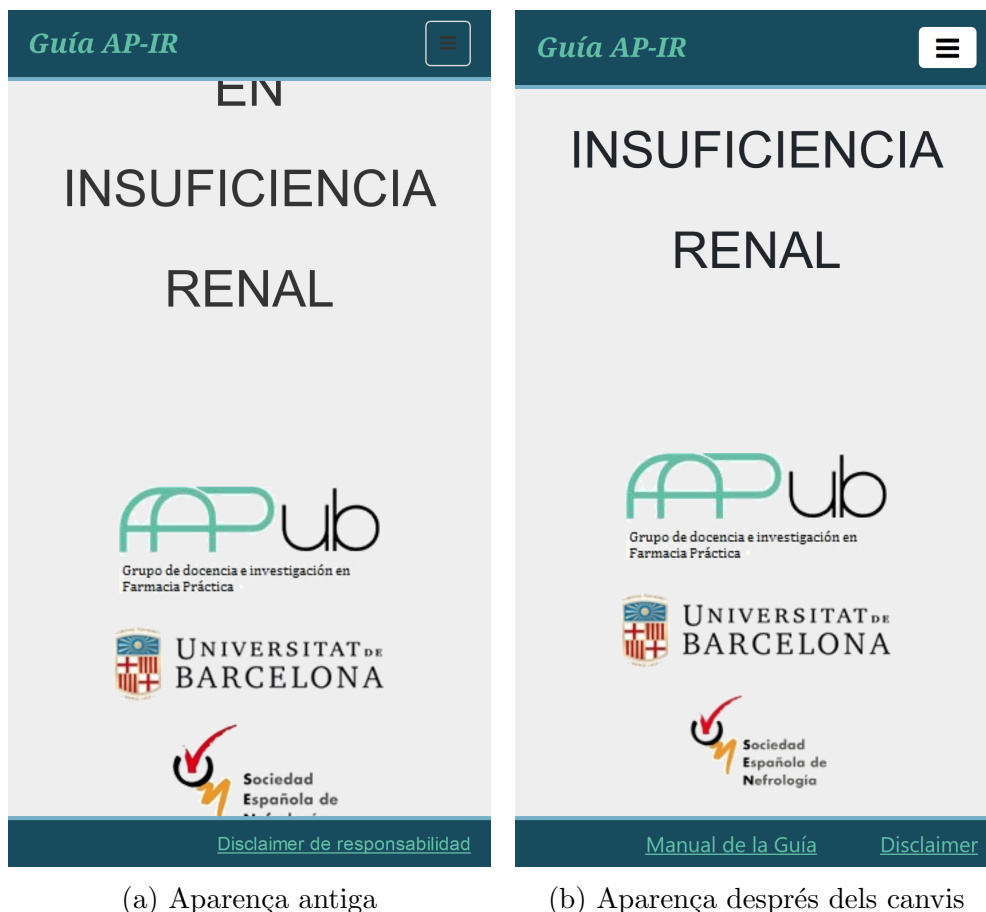


Figura 12: El footer tapava els elements de la part inferior de la pantalla i el botó de menú desplegable no es visualitzava correctament

L'estructura de la pàgina de registre es la mateixa que la del modal. S'ha afegit un botó per eliminar la data de naixement i missatges de feedback, si hi ha hagut un error es mostra a la mateixa pàgina i si tot ha anat bé es mostra un missatge després de redirigir a la pàgina principal. Els missatges es mostren per mitja de alerts de Bootstrap. Quan es registra l'usuari rep un mail amb un enllaç per activar la seva compta. Aquest enllaç porta a una pàgina buida on per mitjà de Javascript es crida al mètode de la Api per activar la conta i després es redirigeix a la pàgina principal mostrant un missatge indicant si ha anat bé o no. Al activar la conta no es pot autenticar automàticament a l'usuari degut a que no disposam de la contrasenya sense xifrar, totes les contrasenyes es xifren per mitjà d'un algorisme unidireccional que fa impossible recuperar la contrasenya original a partir del hash.

La pàgina d'autenticació té la mateixa estructura que el modal. El que si que s'ha canviat son els missatges d'error. Ara en comptes d'informar quina de les dues credencials és la incorrecta tan sols informa de que no són correctes, una petita millora de seguretat. Quan l'usuari s'autentica correctament es redirigit a la pàgina principal però no es mostra cap missatge de feedback ja que es considera prou feedback l'aparició de nous enllaços accessibles a la navbar i al fet que s'ocultin els enllaços a les pàgines d'autenticació i registre. També a la pàgina d'autenticació, tenim un enllaç per recuperar la contrasenya si l'usuari l'ha oblidada, això provoca que s'envii un correu amb l'enllaç al formulari de canvi de contrasenya. El token de canvi de contrasenya només funciona un sol cop així es redueix el risc de que pugui ser utilitzat per fins malintencionats.

Finalment s'ha arreglat un problema que provocava que el footer es superposés sobre els elements de la part més inferior de la pàgina i s'ha afegit un nou enllaç, aquest permet accedir a la pàgina d'informació de la guia.

Una volta autenticat l'usuari guanya accés a les pàgines de consulta i de visualització de les dades de la compta. Si no és l'administrador de la pàgina a més guanya accés a la pàgina d'enviar suggeriments, en canvi si es administrador té accés a la visualització de tots els suggeriments rebuts, a la pàgina de visualització de tots els usuaris i a la de resolució de conflictes, principis actius duplicats i errors de rangs de risc superposats. Aquesta darrera pàgina és una nova funcionalitat que ha sorgit al considerar que la resolució d'aquests conflictes per mitjà d'un mètode automàtic podria portar a la pèrdua de informació important. S'ha eliminat la pàgina que permetia restaurar la base de dades a partir d'un Excel i eliminar tota la base de dades ja que eren funcionalitats que ja no s'empraven i que constituïen un risc per a la seguretat de les dades.

La pàgina de consulta permet cercar un principi actiu i filtrar les recomanacions disponibles en funció del filtratge glomerular. Un del principals canvis es que ja no es mostren els principis actius per defecte ja que una de les funcionalitat que s'havia d'implementar era guardar els principis actius que cercaven els usuaris i si es mostren principis actius per defecte no podríem reunir dades sobre aquests ja

que no l'usuari no utilitzaria la funció de cerca per aquests. A més ara només es fa la cerca al clicar damunt el botó de cerca. Es un pas extra que obligam a fer a l'usuari però així podem controlar millor quins principis actius cerca l'usuari. A més l'usuari administrador té accés a les funcionalitats d'afegir, editar i eliminar principis actius i recomanacions. A diferència d'abans ara es necessari clicar damunt el botó de modificar per accedir a la pàgina on es mostra el formulari d'edició del principi actiu i les seves recomanacions. D'aquesta forma l'usuari administrador pot emprar la pàgina de consultes com si fos un usuari normal i no preocupar-se de modificar alguna cosa per error. A més d'aquesta forma els formularis d'afegir i modificar principi actiu son iguals. En aquesta pàgina es troba l'únic modal de la nova versió de la web, el que abans era el pop-up d'explicació del FGe, que en dispositius mòbils no s'acabava de veure prou bé i que a més a vegades no desapareixia tapant el camp del filtratge ara es un modal que en dispositius mòbils es veu molt millor al quedar centrat al mig de la pantalla. Finalment, s'han eliminat tots els pop-up de la pàgina que s'han convertit en petits missatges d'ajuda mostrats a davall dels camps corresponents, el motiu d'aquesta decisió recau en que encara que un pop-up dona més visibilitat normalment no acaba funcionant correctament en dispositius mòbils. El darrer canvi que es pot apreciar es que el missatge d'avís del camp de FGe ara només es mostra després d'interactuar amb el camp si és necessari.

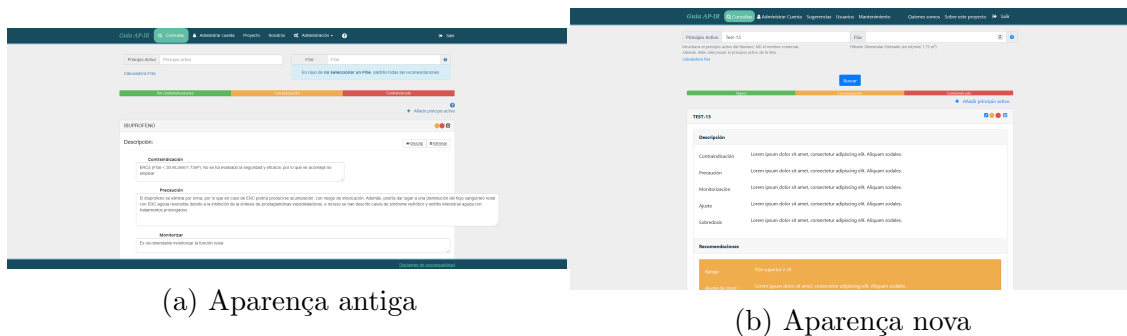


Figura 13: Comparativa de la pàgina de consulta

La pàgina de visualització de les dades de la compta ha quedat igual que abans per a l'usuari no administrador; tan sols s'ha modificat una mica l'aspecte visual degut al canvi de versió de Bootstrap. Per a l'usuari administrador s'ha afegit la funcionalitat de baixar les taules de principis actius, recomanacions i registres de cerca en format CSV.

La pàgina d'enviar suggeriments es nova, ara per enviar un suggeriment ja no s'ha d'enviar des del boto que apareixia als principis actius, de fet aquest boto ja no existeix. S'ha decidit posar-ho com enllaç a la barra de navegació i no com un botó perquè no quedava de quedar clar que el botó servia per enviar suggeriments i a més es trobava una mica amagat. Un altre canvi que s'ha aplicat als suggeriments es el fet que ara es guarda quan s'ha enviat i que ja no es obligatori enllaçar el suggeriment a un principi actiu.

S'ha modificat la pàgina de veure suggeriments, sobretot l'aparença visual i s'ha afegit l'opció d'eliminar els suggeriments un a un. A més tant a aquesta com a la pàgina de visualització d'usuaris s'ha aplicat paginació a la part de servidor per reduir el temps d'espera que a la pàgina d'usuari pujava a uns 17-18 segons. D'aquesta darrera també s'ha canviat l'aparença visual ja que les taules html tenen el problema que no és gens senzill fer-les responsive, a més es va decidir que la taula aportava molts més detalls dels que realment es necessitaven mostrar dels usuaris.

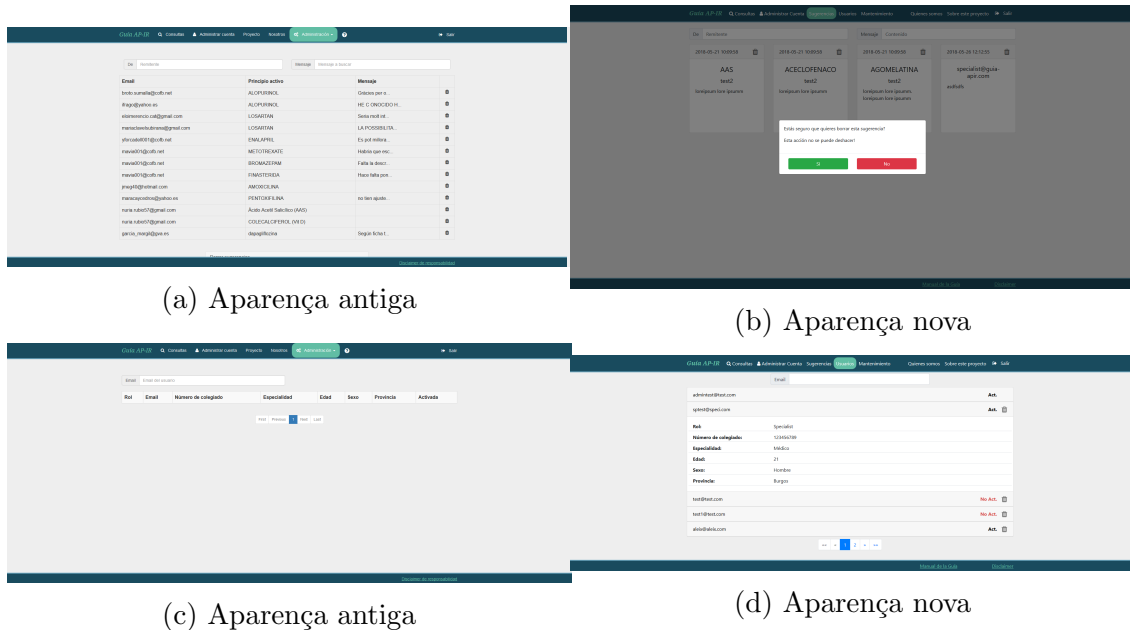


Figura 14: Comparativa de la pàgina de suggeriments i usuaris

Detalls de la implementació

A continuació s'explicaran certs detalls de la implementació de la pàgina amb Angular.

Es començarà pels formularis. Angular disposa de dues formes per implementar formularis, basat en template i reactius. La diferencia principal entre els dos es la forma en que son gestionats. Al formulari basat en template, Angular crea un objecte formulari i l'usuari només s'ha d'encarregar de enllaçar als valors dels camps, les variables definides en el component. En canvi, en els formularis reactius, l'objecte formulari es creat i controlat directament per l'usuari i aquest es enllaçat als diferents camps definits en el template. A l'aplicació s'han utilitzat majoritàriament els formularis reactius principalment degut al fet que ens evita la creació de nombroses variables per enllaçar als camps, o, en el seu defecte, un objecte contenidor de dades per cada formulari. També permet una major personalització del formulari i afegir o eliminar camps dinàmicament. Una aplicació d'aquesta darrera propietat s'ha utilitzat per les pàgines d'edició i creació de principis actius i de pacients. En les pri-

meres per afegir o eliminar recomanacions dels formularis i en les segones per afegir o eliminar tractaments. A continuació es pot veure part del formulari de creació de principi actiu juntament amb un petit extracte del codi Typescript que ho sustenta.

The screenshot shows a web interface with a dark blue header containing navigation links: 'Guía AP-IR', 'Consultas', 'Administrar Cuenta', 'Sugerencias', 'Usuarios', 'Mantenimiento', 'Quiénes somos', 'Sobre este proyecto', and 'Salir'. Below the header, there's a light gray sidebar with the text 'Sobre recomen...' and a main content area. The main area has a title 'Recomendaciones' and a form with the following fields: 'Rango' (a range input from 0 to 140), 'Riesgo' (a dropdown menu), 'Ajuste de dosis' (a large text area), and 'Seguimiento' (another large text area). At the bottom of the form are two blue buttons: 'Añadir recomendación' and 'Guardar cambios'. The footer of the page contains links for 'Manual de la Guía' and 'Disclaimer'.

Figura 15: Formulari de creació de principi actiu: Recomanacions

```
export class AddActivePrincipleComponent implements OnInit {
  ...
  activePrincipleForm: FormGroup;
  ...

  constructor( ...
    private fb: FormBuilder) {
    ...
    this.createForm();
    ...
  }

  createForm() {
    this.activePrincipleForm = this.fb.group({
      principio_activo : ['', Validators.required ],
      contraindicacion: '',
      precaucion: '',
      monitorizacion: '',
      ajustes: '',
      sobredosificacion: '',
      recomendations: this.fb.array([]),
    });
  }
  ...
}
```

```

buildFormGroupArray(recomendations: Recommendation[]) {
  if (recomendations.length === 0) {
    return [];
  }
  return recomendations.map(recomendation =>
    this.buildRecomendationForm(recomendation));
}

buildRecomendationForm(recomendation: Recommendation): FormGroup {
  return this.fb.group({
    riesgo: [recomendation.riesgo, Validators.required],
    fge_min: ...,
    fge_max: ...,
    ajuste_dosis: recomendation.ajuste_dosis,
    seguimiento_clinico: recomendation.seguimiento_clinico,
  });
}
...
addRecomendation() {
  this.recomendations.push(this.buildRecomendationForm(new Recommendation()));
}
...
}

```

Angular proveeix d'una classe factoria, FormBuilder, per a crear l'objecte formulari, FormGroup. En aquest cas definim diversos camps, FormControl i una llista de formularis, FormArray. Aquesta llista no és més que un conjunt d'objectes FormGroup. Per a cada FormControl es pot definir el valor inicial, les regles de validació que s'han d'aplicar, quan s'han d'actualitzar els canvis, entre d'altres. El FormArray, més un ngFor al template, es el que ens permet afegir i eliminar els formularis de recomanacions.

Per a fer les crides Ajax al servidor s'utilitza l'objecte HttpClient que ofereix Angular. HttpClient retorna un objecte de tipus Observable, és a dir, un objecte que retornarà un valor quan acabi la crida asíncrona. La particularitat de l'HttpClient es que retorna el que s'anomena Observable fred, és a dir, que la crida Ajax no s'executa fins que no es subscriu algú a l'Observable. Un altre efecte a tenir en compte referent d'això es que s'executa la crida cada cop que hi ha una subscripció a l'objecte. Per tant, si es necessita fer una subscripció en dos llocs diferents a una mateixa crida Ajax s'han d'utilitzar els mètodes de RxJs. Concretament publishLast i refCount, el primer crea un nou Observable que retransmet el darrer valor de l'original, mentre que el segon es subscriu a l'original quan rep la primera subscripció a l'Observable generat per publishLast i es de-subscriu quan s'ha de-subscrit la darrera subscripció al nou Observable. Al següent extracte de codi es pot veure l'ús de publishLast i refCount per a tenir dues subscripcions a un mateix Observable i fer només una crida Ajax.

```
return this.httpClient.post<LoginResponse>(this.login_url, form).pipe(
publishLast(),
refCount(),
tap(data => {...})
);
```

8.2 Aplicació Lumen

Estructura

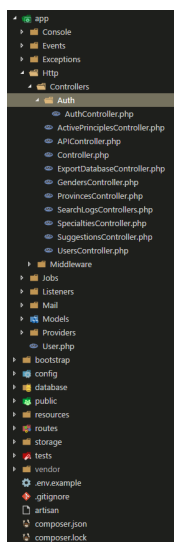


Figura 16: Estructura de l'aplicació Lumen

De totes les carpetes que crea Composer al muntar l'esquelet de l'aplicació Lumen tan sols interessen les següents. En primer lloc, controllers, situada dins app/http. Aquesta carpeta conté els arxius PHP que s'executen en funció de la url de la REST API que se cridi. S'ha creat un arxiu per a cada un dels models de l'aplicació que es recuperen des de l'aplicació, Users, Recommendations, ...; també s'han creat arxius per certes funcions específiques de la API, com l'autenticació i l'exportació de la base de dades a CSV.

Seguim amb la carpeta de middleware també dins Http. Dins aquesta es situen funcions que s'executaran abans d'executar la funció associada a la ruta si es compleixen certes condicions definides a l'arxiu de rutes. S'han implementat tres middlewares; el primer d'ells permet el CORS, Cross-Origin Resource Sharing, que permet o bloqueja peticions fetes a servidors situats en un domini diferent al de la pàgina que fa la petició. Això s'aconsegueix mitjançant l'ús del mètode HTTP Options i Headers especials que indiquen si el servidor permet l'accés al recurs

a l'origen. El segon middleware implementat habilita l'autenticació per mitjà de JWT, el que ens permet restringir certs recursos del servidor a usuaris registrats. Finalment, el tercer middleware és l'encarregat de restringir l'accés als recursos pels que es necessita tenir el rol d'administrador per a poder accedir-hi.

La carpeta de models conté els arxius PHP que defineixen el objectes que Eloquent s'encarregarà de enllaçar al model de dades guardat en MySQL. Per a cada objecte es defineixen les columnes que poden ser assignades per mitjà del mètode create. També es permet definir les columnes que no es vol que es mostrin a la representació JSON del recurs, útil per a no enviar al client dades sensibles com poden ser el token d'activació, el de recuperar contrasenya o la contrasenya en el cas de l'usuari. Aquesta representació JSON també pot ser alterada mitjançant l'ús de mètodes que es criden al intentar recuperar un atribut del recurs, coneguts com

Mutators. Aquesta funcionalitat s'ha utilitzat per a traduir les claus foranes a les descripció del recurs associada a dita clau. Això ens permet, per exemple, enviar el nom del rol en comptes de l'identificador.

Seguim amb la carpeta de Mail. En aquesta carpeta es defineixen les classes encarregades de definir i construir l'objecte Mailable que es processarà com un correu electrònic. Aquest es un canvi introduït a les version més noves de Laravel, ja que abans els correus s'enviaven amb la funció send indicant el destinatari, el template i les dades que s'havien de passar al template. A més, com que Lumen no suporta l'enviament de correus per defecte, es va haver d'importar el paquet de Mail de Laravel i configurar-lo per fer-lo funcionar amb Lumen.

La carpeta config conté els arxius de configuració necessaris per a configurar les llibreries utilitzades en aquest projecte. Aquesta es una carpeta que Lumen no crea al no necessitar-la, només es crea a les aplicacions Laravel. L'avantatge que té Lumen es que, al estar basat en el nucli de Laravel, la majoria de llibreries compatibles amb Laravel també funcionen a Lumen afegint l'arxiu de configuració i unes poques línies a l'arxiu app.php de la carpeta bootstrap.

La carpeta bootstrap mencionada anteriorment conté l'arxiu més important de tota l'aplicació Lumen ja que es l'arxiu encarregat de muntar i desplegar tota l'arquitectura de la mateixa. Entre d'altres carrega les rutes a les classes utilitzades a l'aplicació, llegeix les variables d'entorn des de l'arxiu .env, enllaça els middlewares i les rutes de la API. També registra els proveïdors de serveis, les facades i les configuracions necessàries.

A database hi trobem els seeders, que permeten emplenar la base de dades amb dades falses així com els arxius de migracions per a aixecar tota l'estructura de base de dades definida per a l'aplicació.

Public conté l'index.php i l'arxiu .htaccess. Aquests dos arxius també són de vital importància per a l'aplicació. El primer es l'encarregat de muntar l'aplicació cridant a app.php de la carpeta bootstrap i d'enllaçar la petició entrant al mètode corresponent del controller segons estigui definit a l'arxiu de rutes. El segon s'encarrega de capturar tota petició que arribi i reescriure-la per tal de que sigui tractada per l'index.php. Aquest dos arxius actuant en conjunt són el que fa possible que les rutes definides funcionin.

Per acabar queden dues carpetes, routes i resources. La primera conté l'arxiu on es defineixen les diferents rutes de la API. El segon conté les template en format blade dels correus electrònics d'activació de la compta i recuperació de la contrasenya.

API REST

Per al projecte s'ha recreat la API REST de zero. El primer que es nota es que ha canviat totalment la forma d'autenticació, de bàsic a JWT. A més s'ha intentat utilitzar correctament els mètodes http per a seguir la filosofia rest, GET per a recuperar, POST per a crear, PUT o PATCH per actualitzar i DELETE per eliminar recursos. A més els endpoints de la api estan en plural a excepció de les funcionalitats no associades a recursos com l'autenticació, l'usuari registrat (/me) i l'exportació de la base de dades a CSV.

Les funcionalitats de la api rest s'han distribuït de la següent forma. En primer lloc, funcionalitats públiques que no requereixen de token per accedir-hi. En segon lloc, funcionalitats privades que requereixen la presència del header Authorization i un token vàlid per poder accedir-hi. Finalment, les funcionalitats d'administrador, que a més del token vàlid requereixen que l'usuari al que pertany el token tingui el rol d'administrador.

Les funcionalitats públiques comprenen el mètode d'autenticació i refresc de token, el registre d'usuari així com l'accés als recursos d'especialitat, gènere i província, al ser requerits per al registre d'usuari. També inclou els mètodes relacionats amb la pèrdua i restauració de contrasenya a més del mètode per activar la compta d'usuari. El mètode d'autenticació utilitzat és JWT. Per obtenir aquest token cal proporcionar les credencials, correu electrònic i contrasenya. Si les credencials són vàlides el servidor retorna un token amb una validesa de 1 hora i que pot ser refrescat durant dues setmanes. Al cap de les dues setmanes és obligatori tornar a obtenir un token proporcionant les credencials. El registre d'usuari accepta una crida POST, és comprova que les dades proporcionades siguin correctes comprovant que el correu electrònic proporcionat no estigui ja registrat, que les contrasenyes tinguin la llargària mínima i coincideixin entre altres. Finalment, si tot es correcte, es crea l'usuari, generant un token d'activació i enviant un correu electrònic a l'usuari amb l'enllaç per activar la contrasenya. El mètode de recuperar contrasenya es comporta de forma similar, comprova que el correu estigui registrat, genera un token de recuperació i envia un link a l'usuari per a accedir a la pàgina de canvi de contrasenya. Pel que fa als recursos accessibles, aquests només suporten el mètode GET, és a dir, son de només lectura.

Pel que fa a les funcionalitats privades, en primer lloc tenim la funcionalitat de recuperar les dades de l'usuari autenticat. Aquest s'obté a partir del token proporcionat. A continuació tenim la funcionalitat de donar de baixa la conta d'usuari. En aquest mètode s'ha implementat la comprovació de que l'usuari no sigui un administrador. Encara que a la pàgina web no hi hagi forma de cridar aquest mètode, al ser una api, un usuari malintencionat que hagi obtingut el token de l'administrador podria fer una crida a aquesta ruta i esborrar la conta. També es troba el mètode de canvi de contrasenya, que com es d'esperar, permet fer el canvi de contrasenya de l'usuari autenticat. Tot seguit arribam a la ruta que proporciona

la funcionalitat principal de l'aplicació web, la cerca de principis actius. Aquest mètode accepta un `get` però requereix que se li passi com a paràmetre de Query un String de cerca de com a mínim 4 caràcters. Aquesta limitació ha estat demanada per l'equip de farmàcia per així evitar que puguin clonar fàcilment la informació sobre aquests principis actius. Aquesta crida retorna les dades dels principis actius juntament amb les recomanacions associades si les tenen. Per acabar, en aquest grup també trobem les funcionalitats de guardar els registres de cerca i els suggeriments. Ambdós mètodes accepten un `POST`, com és comú en totes les API REST.

Finalment arribam als mètodes d'administrador, els quals comprenen el gruix de les funcionalitats CRUD de l'aplicació. Primer de tot trobam l'enllaç per a recuperar tots els usuaris. Aquest retorna mètode genera els enllaços necessaris per a proporcionar una paginació a la part de servidor. D'aquesta forma evitam els quasi 20 segons d'espera que tenia l'anterior pàgina web. També hi ha el nou mètode per eliminar qualsevol usuari, que com el de donar de baixa la compta comprova que l'usuari que s'intenta eliminar no sigui un administrador. Trobem els mateixos mètodes per als suggeriments, paginació com a forma de recuperació del recurs i eliminació d'un suggeriment. Tot seguit arribam als mètodes CRUD dels principis actius, creació, actualització i eliminació. Tant a la creació com a l'actualització es comprova que el principi actiu no estigui repetit, que el seu nom tingui la mida mínima necessària i que les recomanacions associades no tinguin rangs de risc superposats. A l'eliminació s'ha d'eliminar primer les recomanacions i registres de cerca associats al principi actiu. També estan definits les rutes per a fer l'exportació a CSV de la taula de principis actius així com els mètodes de recuperació de tots aquells principis actius que estiguin duplicats o que tinguin recomanacions amb rangs de risc superposats. Aquests dos mètodes són els que envien els principis actius conflictius a la pàgina de manteniment de la base de dades per a que siguin resolts. Finalment, tenim les rutes per a exportar les taules de recomanacions i registres de cerca a arxius CSV.

Implementació

En aquest capítol es comentaran tots aquells detalls de la implementació de l'aplicació Lumen que es considerin prou interessants. Es començarà explicant el sistema de validació de les peticions, seguit per l'ORM, Object Relational Mapping, Eloquent, les migracions implementades a la base de dades i, finalment, el sistema d'enviament de correus electrònics.

Per a validar les dades de les peticions s'ha utilitzat el mètode `validate` de Lumen. Aquest mètode permet indicar per a cada paràmetre de la petició tot un seguit de regles. Lumen proporciona un conjunt bastant ampli de regles, 59 en total, que cobreix la casi totalitat dels casos d'ús que puguem necessitar per a validar qualsevol petició entrant. A més, crear noves regles és bastant senzill i a la documentació oficial de Laravel trobam exemples per a implementar-les. Un altre dels avantatges d'aquests mètodes és l'excepció de validació que genera quan no es compleix al-

guna de les regles establertes permet crear la resposta en format JSON que retornar al client. Al següent extracte de codi es pot veure el funcionament del mètode. Rep com a paràmetres l'objecte a validar i un diccionari relacionat cada paràmetre a validar amb el conjunt de regles a aplicar. Les regles es poden indicar com un string o com una llista.

```
<?php
public function postUser(Request $request) {
    try {
        $this->validate($request, [
            'email' => 'required|email|max:255|unique:users',
            'password' => 'required|string|min:8',
            'confirm_password' => 'required|string|min:8|same:password',
            'specialty_id' => 'required|numeric|exists:specialties,id',
            'first_name' => 'required|string|max:255',
            'last_name' => 'required|string|max:255',
        ]);
    } catch (ValidationException $e) {
        return $e->getResponse();
    }
}
```

Per accedir a les dades guardades en base de dades s'ha utilitzat l'ORM que proporciona Lumen, Eloquent. S'han reutilitzat els models que ja havia definit el Maikel modificant tan sols les classes que fossin necessàries. Un dels avantatges d'Eloquent es que permet transformar les dades a format JSON fàcilment. A més permet definir quines dades s'adjunten i quines no durant la transformació. També permet modificar com s'accedeix a cada un dels atributs, el que es coneix com Mutator. Per a definir un Mutator es tan fàcil com definir un nou mètode públic 'get' més el nom de l'atribut en format CamelCase. Al següent extracte de codi es pot veure la implementació d'un dels Mutator del model de l'usuari.

```
<?php
public function getRoleNameAttribute($value) {
    return $this->getDescription($this->role);
}
```

Per a aquest projecte s'han implementat més de 12 migracions diferents. Per a cada una d'elles s'ha de definir el mètode up, que defineix el que ha de fer la migració quan s'apliqui; i el mètode down, que defineix les modificacions necessàries a la base de dades per a desfer la migració. S'ha intentat reduir al mínim les modificacions que apliquen cada una de les migracions per a obtenir arxius més petits i així tenir una major granularitat de les mateixes. Això permet que les migracions siguin més mantenibles. Per a definir les migracions s'ha utilitzat directament el Query Builder i les façanes Schema i Blueprint. Amb el primer s'han definit aquelles

migracions que modifiquen les dades guardades a la base de dades, mentre que les segones defineixen les migracions que alteren el propi model de dades. A l'extracte de codi següent es pot veure un exemple d'ús de cada un d'aquests dos mètodes.

```
<?php
// Modificació del model de dades amb Schema i Blueprint
Schema::table('active_principles', function (Blueprint $table) {
    $table->renameColumn('monitorizacion_si', 'monitorizacion');
    $table->renameColumn('ajustes_si', 'ajustes');
});

// Modificació de les dades de la base de dades amb el Query Builder
$table = DB::table('recommendations');
$table->where('fge_max', '>', 140)->update(['fge_max' => 140]);
```

Per a l'enviament de correus electrònics a Lumen primer s'han d'importar les llibreries de Laravel de Mail i copiar l'arxiu de configuració a la carpeta config. A més, per a utilitzar el driver smtp cal importar la llibreria SwiftMailer. Una volta resolt l'apartat de configuració s'han de crear les classes Mailable que gestionen la creació de l'objecte i la representació de la plantilla a més d'encarregar-se de passar-li les dades necessàries per a la seva interpolació. A continuació es pot veure el codi d'una d'aquestes classes Mailable.

```
<?php
class SendActivationMail extends Mailable
{
    use Queueable, SerializesModels;
    public $base_url;
    public $activation_key;
    public $id;
    /**
     * Create a new message instance.
     *
     * @return void
     */
    public function __construct($activation_key, $id)
    {
        $this->activation_key = $activation_key;
        $this->id = $id;
        $this->base_url = env('LOCAL_URL', "http://www.ub.edu/medicamen...");
    }
    /**
     * Build the message.
     */
}
```



```

    * @return $this
    */
    public function build()
    {
        return $this->subject("Activación de la cuenta")->view(...);
    }
}

```

8.3 Integració de les dues parts

Abans de fer la integració cal compilar i muntar l'aplicació Angular. Això ajunta, i minifica tot el codi Javascript definit a l'aplicació en un sol arxiu. El resultat es una carpeta amb poc més de 8n arxius que conté tota l'aplicació Angular. Aquest arxius simplement s'han de copiar al directori public del servidor apache per a ajuntar les dues parts. La integració de les dues parts ha estat bastant més senzilla del que pareixia en un primer moment gràcies al fitxer .htaccess del servidor apache. Aquest fitxer permet definir unes regles de sobreescritura de URL a aplicar a les peticions entrants. D'aquesta forma podem indicar-li al servidor que ens redirigeixi les rutes que comencin per api a l'arxiu index.php i la resta que les deixi passar a l'arxiu index.html de l'aplicació Angular.

8.4 Pàgina de pacients

Primer s'explicaran el desenvolupament realitzat a la part de servidor, parlant primer de la base de dades i a continuació dels nous endpoints a la REST Api Lumen. A continuació s'explicarà la part de client.

Part de servidor

Pel que fa a la part de base de dades, s'han afegit set noves taules.

1. Taula de pacients:

Taula que guarda les dades bàsiques dels pacients. La informació a guardar s'ha decidit a partir de les taules proporcionades per l'equip de farmàcia. Conté les següents columnes:

- user_id: Identificador de l'usuari que ha introduït el pacient.
- created_at i updated_at: Timestamps inserits automàticament per Eloquent.
- gender_id: Clau forana que apunta a la taula genders.
- province_id: Clau forana que apunta a la taula provinces.

- fge: Nivell de FGe del pacient.
 - imc: Index de Massa Corporal del pacient.
 - age: Edat del pacient.
 - alergies: Al·lèrgies que sofreix el pacient, si en té cap que sigui rellevant.
 - biologic_params: Paràmetres Biològics, camp de text.
 - identifier: Identificador de 8 caràcters creat automàticament pel servidor al crear el pacient.
2. Taula de problemes de salut:
- Taula que relaciona un problema de salut amb un pacient determinat, si el pacient es elimina tots els problemes de salut relacionats amb aquest també s'eliminen. Conté les següents columnes:
- paciente_id: Clau forana que indica el pacient al qual pertany aquest problema de salut.
 - inicio: Data en la qual va començar a sofrir el problema de salut el pacient.
 - description: Breu descripció del problema de salut.
3. Taula de Tractaments:
- Taula que relaciona un tractament amb un problema de salut determinat, si el problema de salut es elimina el tractament s'elimina automàticament. Conté les següents columnes:
- ps_id: Problema de salut que s'està tractant amb el principi actiu del tractament.
 - inicio: Data en la que el pacient va començar a prendre el tractament.
 - active_principle: Nom del principi actiu amb el qual s'està tractant el problema de salut.
 - pauta_preescrita: Pauta de dosis prescrita pel metge.
 - pauta_usada: Pauta de dosis que està prenent realment el pacient.
4. Taula de RNM:
- Taula que relaciona un RNM amb un problema de salut determinat, si el problema de salut es elimina el RNM s'elimina automàticament. Conté les següents columnes:
- ps_id: Problema de salut que presenta un RNM degut a algun dels medicaments utilitzats per tractar-lo.
 - control: Camp de text.
 - preocupa: Indica si el pacient està preocupat o no pel problema de salut.
5. Taula de PRMS:
- Taula que relaciona un PRM d'un tractament d'un problema de salut amb el RNM del mateix problema de salut, com els altres si el RNM es elimina aquest també s'elimina automàticament. Conté les següents columnes:

- `rnm_id`: Clau forana que apunta al RNM relacionat amb el PRM.
- `tratamiento_id`: Clau forana que apunta al tractament relacionat amb el PRM.
- `riesgo`: Nivell de risc que presenta el PRM per al pacient.
- `necesario`: Indica si el principi actiu es necessari per a tractar el problema de salut.
- `efectivo`: Indica si el principi actiu es efectiu per a tractar el problema de salut.
- `seguro`: Indica si el principi actiu es segur per a tractar el problema de salut.
- `prm_type`: Tipus de PRM per la pràctica en Insuficiència Renal. Aquests son Contraindicació i Dosis, pauta i/o duració no adequada.
- `prm_description`: Breu descripció del PRM

6. Taula d'intervencions:

Taula que conté les diferents intervencions que es poden aplicar a un PRM per a fer-lo segur per al pacient.

7. Taula auxiliar d'intervencions i prms:

Aquesta taula relaciona cada PRM amb el conjunt d'intervencions aplicades.

S'ha dissenyat la taula de tal forma que al eliminar un pacient totes les taules que depenen d'aquesta, és a dir, totes a excepció de la d'intervencions eliminin les files que estaven relacionades amb aquest pacient. S'ha fet així per a assegurar la consistència de les dades ja que aquestes files només poden existir si existeix el pacient.

Pel que fa a la part de la REST Api, s'han afegit dos nous endpoints. El primer el de pacients s'encarregà de permetre recuperar la llista de pacients i recuperar, editar i eliminar un pacient concret. Per a recuperar, editar i eliminar un pacient concret es necessari proporcionar l'identificador del pacient i després es comprova que l'usuari tingui assignat el pacient que està intentant recuperar o eliminar. Per a introduir un pacient s'ha fet un ús intensiu de les funcionalitats d'Eloquent per a guardar automàticament les relacions entre les diferents taules, la qual cosa ha facilitat bastant el desenvolupament. En el cas de l'edició d'un pacient primer s'actualitzen les dades de les columnes de la taula pacient i tot seguit s'eliminen tots el problemes actius del pacient per a ser reintroduïts de nou. S'ha decidit fer així ja que simplifica l'actualització de les dades al no tenir de preocupar-se de comprovar quines dades s'han eliminat i quines no. Encara que també provoca que editar el pacient sigui una tasca costosa independentment de que tan petit sigui la modificació feta per l'usuari. El segon endpoint recupera la llista d'intervencions.

8.5 Part de client

Aquesta ha estat la part més complicada d'aquesta nova pàgina. El primer que s'ha fet ha estat crear dos nous serveis que fan les crides als nous endpoints de la REST Api.

Tot seguit s'ha procedit a crear la vista del llistat de pacients, aquesta s'ha basat en la llista d'usuaris però degut a la gran quantitat d'informació a mostrar s'ha decidit la creació d'una vista detallada del pacient. Així al desplegar un pacient simplement es mostren els detalls bàsics i es disposa d'un botó per anar a la vista detallada. En aquesta vista es veuen dues pestanyes, a la primera es mostren les dades corresponents a la taula d'Estat de Situació proporcionada per l'equip de la Facultat de Farmàcia, a la segona es mostren les dades corresponent a la taula d'Avaluació juntament amb les diferents intervencions de cada PRM.

[illegible]

(a) Taula d'Estat de Situació

[illegible]

(b) Taula d'Avaluació

Figura 17: Taules proporcionades per l'equip de la Facultat de Farmàcia

Una volta s'ha tingut aquesta part completada s'ha procedit a crear el formulari de creació de pacients. Aquest formulari hauria de servir també per a editar un pacient. Per a aconseguir el resultat esperat s'ha decidit utilitzar l'únic recurs que té Angular per a crear un formulari de les característiques exigides per l'estructura de les dades del pacient. Abans però cal explicar el format en que retorna l'objecte la REST Api. L'estructura és la següent, s'han eliminat la majoria d'atributs per a evitar que l'objecte ocupés més d'una pàgina:

```
{
  ...
  "problemas_salud": [
    {
      ...
      "treatments": [
```

```

        {
            ...
        }
    ],
    "rnm":
    {
        ...
        "prms": [
            {
                ...
                "interventions": [...]
            }
        ]
    }
}
]
}

```

El principal problema d'aquesta representació el trobam a l'atribut PRM, que com s'ha dit a l'apartat anterior està relacionat amb el RNM i amb un tractament del problema de salut. Per tant necessitam representar aquesta relació al formulari de pacients. Per aconseguir això cal entendre exactament com funciona el formulari reactiu d'Angular. Aquest ja s'ha explicat a l'apartat 8.1. Per resumir-ho, primer es defineix l'objecte formulari a l'arxiu Typescript juntament amb tots els controls que poden ser assignats o no a un input de frontend. A frontend es relacionen els diferents elements del DOM amb els atributs de l'objecte formulari. Al següent exemple es pot veure el funcionament:

A la part de Typescript es defineix l'objecte:

```

reactiveForm = FormBuilder.group({
    control_1: valor_inicial,
    control_2: valor_inicial
    group: FormBuilder.group({
        sub_control_1: valor_inicial
    })
});

```

I a la part HTML s'enllaça amb els elements del DON;

```

<div [FormGroup]="reactiveForm">
    <input type="text" formControlName="control_1">
    <input type="text" formControlName="control_2">
    <div formGroupName="group">
        <input type="text" formControlName="sub_control_1">
    </div>
</div>

```

Per tant sembla evident que un control de l'objecte s'assigna a un input del template i que segueix la mateixa estructura que la definida a l'objecte. Però Angular no imposa aquesta restricció, és a dir, es podem assignar més d'un input del template a un control de l'objecte. Per exemple, el següent template també seria valid.

```
<div [FormGroup]="reactiveForm">
  <input type="text" formControlName="control_1">
  <input type="text" formControlName="control_1">
  <div formGroupName="group">
    <input type="text" formControlName="sub_control_1">
  </div>
  <input type="text" formControlName="control_1">
  <div formGroupName="group">
    <input type="text" formControlName="sub_control_1">
  </div>
</div>
```

Aquí es pot veure com s'ha assignat més d'un input a alguns controls de l'objecte i fins i tot que un dels control no té assignat cap input. Els diferents inputs assignats a un mateix control tindran el seu valor automàticament sincronitzat per Angular. Aquesta propietat dels formularis reactius es la que s'ha utilitzat per a crear el formulari de pacients.

S'ha estructurat de la mateixa forma que la vista detallada del pacient, en dues pestanyes on a la primera permet introduir les dades corresponents a la taula d'Estat de Situació. És a dir, aquí s'introdueixen les dades bàsiques del pacient, els seus problemes de salut i els tractaments que pren per a cada un d'aquests. Després a la segona pestanya és on es permet introduir els RNM dels problemes de salut.

Per a introduir un nou RNM s'utilitza un desplegable que mostra tots els problemes de salut introduïts que no tinguin ja assignat un RNM, ja que un problema de salut només pot tenir un únic RNM assignat. Si el problema de salut ja té un RNM assignat aleshores apareix directament. Per cada RNM es mostren deshabilitats les dades introduïdes a la pestanya anterior del problema de salut i dels tractaments que tingui assignats. S'ha decidit deshabilitar les dades repetides per a simplificar la validació del formulari. Per tant només es troben habilitats els inputs assignats als controls dels sub-formularis de RNM i PRMs.

Per a aconseguir aquesta estructura de dues pestanyes a l'HTML, a la primera es defineix la llista de tots els FormGroup de Problemes de Salut mentre que a la segona és repeteix però filtrada per tal de mostrar únicament aquells problemes de salut que no tinguin assignat el sub-formulari de RNM.

9 Proves i resultats

L'aplicació s'ha anat provant mentre es desenvolupava. En molts casos s'ha hagut de recórrer a prova i error per aconseguir fer funcionar les diferents parts. Una volta es va tenir l'aplicació acabada es va provar cada part per separat, comprovant que els mètodes de la API funcionassin com s'esperava i que la web reaccions com s'esperava davant les diferents accions de l'usuari.

Les proves més extensives van ser executades damunt el servidor públic de Heroku. En aquest es tenia la pàgina en l'entorn més similar possible al disponible en el servidor de la UB. Es va comprovar que l'aplicació funcionés fent un cicle complert de funcionament, començant amb l'usuari registrant-se i comprovant que arribava el mail d'activació. Fent una recuperació de contrasenya i veure que arribava el correu per accedir a la pàgina de recuperació de contrasenya i que aquesta es canviava correctament. Que l'usuari es podia autenticar i rebia els privilegis correctes; podia cercar principis actius i enviar suggeriments però no podia accedir a les pàgines d'administrador. Després es va provar a autenticar com administrador, comprovant que es mostraven les pàgines que només aquest usuari tenia accés.

Durant aquesta fase de proves es van veure errors que no s'havien detectat durant la fase de desenvolupament. El més greu sent un error que provocava que al fer la restauració de contrasenya s'enviava la petició dues vegades. Això provocava que la primera petició passés, eliminés el token de restauració i al arribar la segona aquesta fallés ja que el token ja s'havia invalidat. El resultat era que a l'usuari se li mostrava un missatge d'error. La resta eren errors bastant senzills de resoldre. Errors de template que apareixien en un cert mida de pantalla, errors degut a l'autocompleter del navegador i finalment que el botó d'exportar a CSV no funcionava degut a que no s'havia pensat a implementar una funció en una de les llibreries. Aquest error va se resolt creant una classe que heretava de la classe de la llibreria i implementava la funció que no estava definida dins la llibreria.

9.1 Pantalles

En aquesta secció es mostraran algunes de les pantalles que encara no s'han mostrat. Començant per la pantalla de consultes al mostrar un principi actiu.

Guía AP-IR

+ Añadir principio activo

TEST-4

Descripción

Contraindicación	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.
Precaución	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.
Monitorización	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.
Ajuste	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.
Sobredosis	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.

Recomendaciones

Rango

FGe superior a 60

Ajuste de dosis

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.

Seguimiento

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.

Rango

FGe inferior a 60

Ajuste de dosis

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.

Seguimiento

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales.

Figura 18: Principi actiu a la pantalla de consultes

Aquí les recomanacions tenen com a fons el color del grau de severitat: segur, amb precaució o contraindicat.

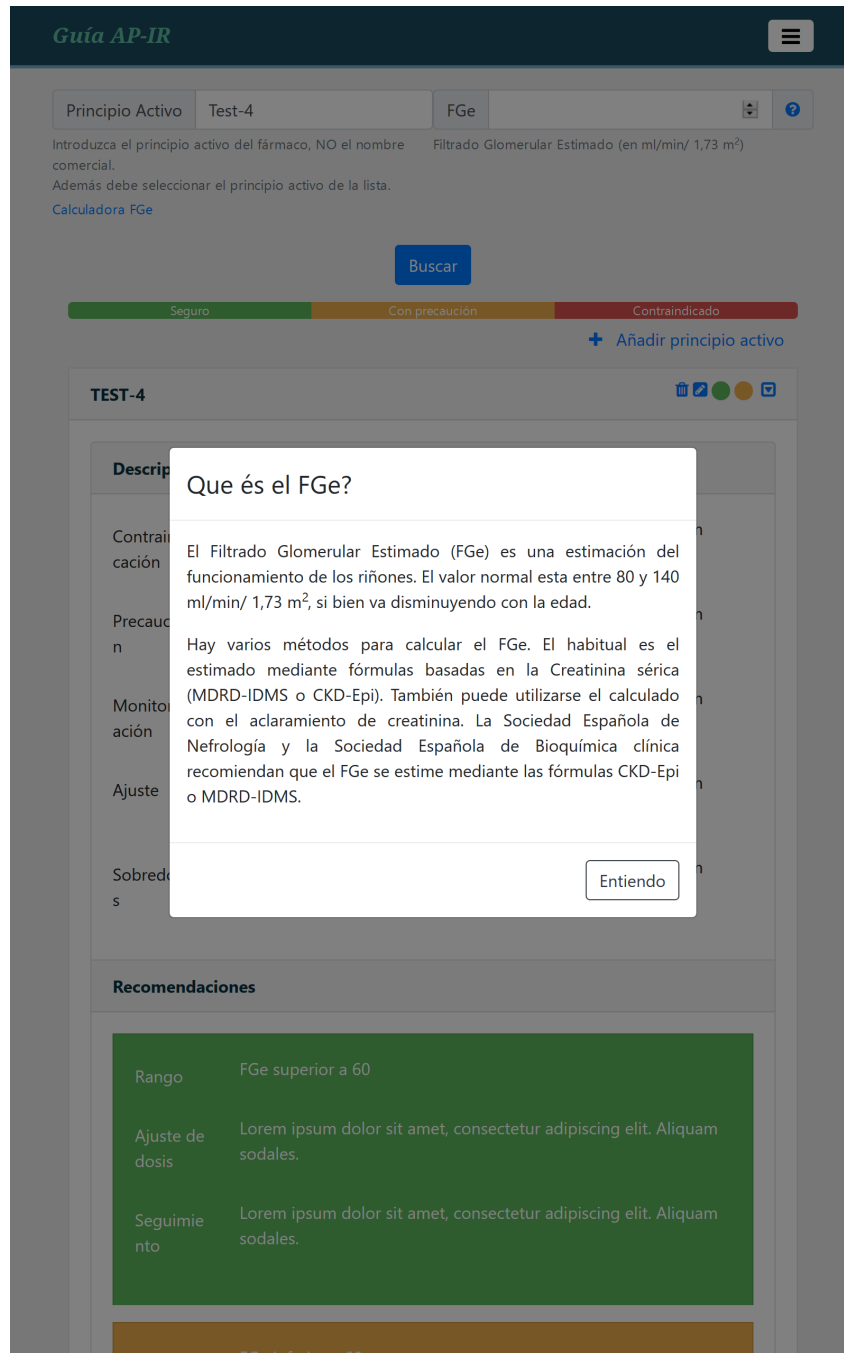


Figura 19: Modal d'informació sobre el FGe

Es va decidir ficar un modal perquè realment era molta informació per a un simple pop-up, així es mostra la informació de forma clara i una volta acabat de llegir simplement es tanca clicant al botó Entendido.

Figura 20: Pantalla de resolució de conflictes

Figura 21: Pantalla de dades de la compta per a l'usuari administrador

L'usuari no administrador, en comptes de tenir el botó de descarregar CSV té el formulari per a cancel·lar la compta.

Figura 22: Pantalla de registre

El botó al costat del selector de data permet esborrar el camp.

Figura 23: Format dels missatges

Els missatges d'error, informació o alerta tenen aquest estil, tan sols canvia el color de fons de l'alerta.

10 Conclusions i treball futur

10.1 Conclusions

S'ha aconseguit complir tots els objectius plantejats al principi del projecte. Un punt a millorar seria la planificació del projecte que es la que més ha fallat.

S'ha deixat el projecte llest per a la quarta fase de desenvolupament poder fer l'anàlisi de les dades emmagatzemades gràcies a la implementació de la nova pàgina de pacients per a permetre que els usuaris, concretament els alumnes de Farmàcia, introdueixin dades dels pacients i a que ara es guarden totes les cerques de principis actius realitzades a la guia. El responsive i rendiment de l'aplicació ha millorat bastant, sobretot al eliminar tots els artefactes que dificultaven l'ús de l'aplicació en dispositius mòbils. A més, les versions de les tecnologies utilitzades són les més actuals a excepció de la de PHP i Lumen al dependre de la primera, ja que depenen de la versió instal·lada al servei d'hostatgeria web de la UB. Un dels principals avantatges d'haver actualitzat Angular és que s'han solucionat molts dels problemes de rendiment que sofria AngularJs, a més el seu sistema de formularis i de crides Ajax es molt més madur que el seu predecessor. El fet que utilitzi Typescript en comptes de Javascript també es un avantatge, ja que no només permet el tipat de les funcions si no que permet fer escriure el codi en una estructura orientada a objectes molt similar a com es fa en Java.

Amb la guia de instal·lació del servidor en local de l'apèndix s'espera poder facilitar la feina als pròxims desenvolupadors. Això els permetrà que hagin de dedicar menys temps a preparar l'entorn per a començar a desenvolupar i per tant puguin optimitzar el seu temps.

10.2 Treball Futur

Tal com s'ha explicat a l'apartat 1.2, aquest projecte encara té prevista una quarta fase de desenvolupament on la idea es fer un anàlisi de dades. Aquesta tercera fase tenia l'objectiu de deixar preparada l'aplicació per començar a recopilar dades per a fer possible aquesta quarta fase. Com que ara es guarden els registres de cerca i la informació de pacients i medicaments introduïts pels usuaris s'estan recopilant tot un conjunt de dades d'ús de l'aplicació que poden propiciar que en un futur es puguin fer estudis bastant interessants sobre els principis actius més utilitzats o quins principis actius son receptats a persones amb un FGe baix.

Referències

- [1] Guia web per a l'ús de medicaments en insuficiència renal
<http://www.ub.edu/medicamentoseninsuficienciarenal/about>
- [2] Quiénes somos, Grupo de docencia e investigación en Farmacia Práctica, Facultad de Farmàcia de la Universitat de Barcelona,
<http://www.ub.edu/farmaciapractica/>
- [3] Via-Sosa MA, Lopes N, March M. – Effectiveness of a drug dosing service provided by community pharmacists in polymedicated elderly patients with renal impairment—a comparative study.
BMC Fam Pract. 2013 Jul 13; 14:96. Doi: 10.1186/1471-2296-14- 96
- [4] Nou conveni entre la UB i la Societat Espanyola de Nefrologia per potenciar l'ús segur dels medicaments en pacients. Notícies de la UB. 17 de febrero de 2014
http://www.ub.edu/web/ub/ca/menu_eines/noticies/2014/02/029.html
- [5] ATC: Structures and principles, WHO Collaborating Centre for Drug Statistics Methodology.
https://www.whocc.no/atc/structure_and_principles/
- [6] Quitaquis Tamay, Michel Dennis - Aplicación web para una guía de medicamentos de la facultad de farmacia.
- [7] Renom Vilardell, Sergi: Guia web per a l'ús de medicaments en insuficiència renal: gestió de la base de dades
<http://hdl.handle.net/2445/119562>
- [8] Universitat de Barcelona: Servei d'hostatgeria web
<http://www.ub.edu/tecnicweb/tag/ftp/>
- [9] JSON Web Token
<https://jwt.io/>
- [10] Wikipedia: Diseño web adaptable
https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable
- [11] Heroku: Cloud Application Platform
<https://www.heroku.com>
- [12] PHP: PHP 5 ChangeLog
<http://php.net/ChangeLog-5.php>
- [13] PHP: Unsupported Branches
<http://php.net/eol.php>
- [14] The MariaDB Foundation – Supporting continuity and open collaboration in the MariaDB ecosystem
<https://mariadb.org/>

- [15] MariaDB vs MySQL: ¿cuál debo elegir?
<https://guiadev.com/mariadb-vs-mysql-cual-debo-elegir/>
- [16] Lumen 5.5: Server Requirements
<https://lumen.laravel.com/docs/5.5#installation>
- [17] Ubuntu: Scale out with Ubuntu Server
<https://www.ubuntu.com/server>
- [18] The Heroku Platform
<https://www.heroku.com/platform>
- [19] Composer: Introduction
<https://getcomposer.org/doc/00-intro.md>
- [20] Laravel News: Announcing Lumen
<https://laravel-news.com/lumen>
- [21] Bootstrap 4: Migrating to v4
<https://getbootstrap.com/docs/4.0/migration/>
- [22] Angular powered Bootstrap
<https://ng-bootstrap.github.io/#/home>
- [23] Angular Features
<https://angular.io/features>
- [24] NodeJs
<https://nodejs.org/es/>
- [25] MDN web docs: export
<https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Sentencias/export#Especificaciones>
- [26] Angular Docs: Component Styles: Style Scope
<https://angular.io/guide/component-styles#style-scope>
- [27] TemplateToaster: Bootstrap 3 vs Bootstrap 4
<https://blog.templatetoaster.com/bootstrap-3-vs-bootstrap-4-migrate-differences/>
- [28] Vue.js: Comparison with Other Frameworks: AngularJS - Runtime Performance
<https://vuejs.org/v2/guide/comparison.html#Runtime-Performance-1>
- [29] Vue.js: Comparison with Other Frameworks: Angular - Runtime Performance
<https://vuejs.org/v2/guide/comparison.html#Runtime-Performance-2>
- [30] Results for js web frameworks benchmark – round 7
<https://www.stefankrause.net/js-frameworks-benchmark7/table.html>

- [31] AngularJS and Angular 2+: a Detailed Comparison
<https://www.sitepoint.com/angularjs-vs-angular/>
- [32] Lumen: The stunningly fast micro-framework by Laravel.
<https://lumen.laravel.com/>
- [33] Laravel: The PHP Framework For Web Artisans
<https://laravel.com/>
- [34] Aerospike: PHP 7 vs. PHP 5: A Performance Comparison
https://www.aerospike.com/blog/php7_php5/
- [35] Wikipedia: Autenticación de acceso básica
https://es.wikipedia.org/wiki/Autenticaci%C3%B3n_de_acceso_b%C3%A1sica
- [36] Laravel 4: Security
<https://laravel.com/docs/4.2/security>
- [37] Portalfarma: Atención Farmacéutica
<http://www.portalfarma.com/Profesionales/DestacadosProfesionales/Paginas/Atencion-Farmaceutica-SFT.aspx>
- [38] XAMPP: An easy to install Apache distribution containing MySQL, PHP, and Perl
<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.2.2/>
- [39] PHP: PHP 7 ChangeLog
<http://php.net/ChangeLog-7.php>

11 Apèndix

11.1 Guia d'instal·lació del servidor local

S'explicarà com instal·lar un servidor local per a córrer la pàgina web en un entorn Windows per a poder començar a desenvolupar.

El primer de tot es instal·lar la versió de XAMPP que tingui la versió de PHP instal·lada al servei d'hostatgeria web de la UB, que en el moment de fer el treball es la 7.0.27. Això seria el més recomanable ja que aleshores s'estalvia un dels passos següents. Per aquest treball no es va tenir en compte i es va instal·lar la darrera versió de XAMPP disponible.

Una volta es té el servidor XAMPP instal·lat fa falta configurar el virtual host. Per això cal afegir el següent a l'arxiu httpd-vhosts.conf situat a dins C:

```
xampp
apache
conf
extra.
```

```
<VirtualHost guia-apir.com:80>
    ServerAdmin guia-apir.com
    DocumentRoot "C:/xampp/htdocs/guia-apir-ub/guia-apir-api/public"
    <Directory "C:/xampp/htdocs/guia-apir-ub/guia-apir-api/public">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Es possible que sigui necessari modificar les rutes als arxius en funció de on hagi extret el codi font. En tot cas el següent es baixar el codi font des del repositori privat de la UB on està guardat. Baixar el codi font de la branca dev. Abans de seguir es necessari baixar i instal·lar Composer.

Extreure el fitxer zip baixat dins la carpeta htdocs de xampp. O millor, fer el git clone a dins d'aquesta. Una volta extret obrir una consola de comandes i anar a la carpeta guia-apir-api i executar la següent comanda:

```
composer update
```

Finalment es necessari crear l'arxiu .env a dins aquesta carpeta, ha de contenir el següent:

```
APP_ENV=local
APP_DEBUG=true
```



```
API_DEBUG=true
APP_KEY=local_app_key

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=guia-apir
DB_USERNAME=root
DB_PASSWORD=

CACHE_DRIVER=file
QUEUE_DRIVER=sync

JWT_SECRET=local_jwt_secret

API_PREFIX=api

MAIL_HOST=
MAIL_PORT=
MAIL_DRIVER=smtp
MAIL_ENCRYPTION=null
MAIL_FROM=
MAIL_USERNAME=
MAIL_PASSWORD=
MAIL_NAME="Guia de ajuste renal"
MAIL_PRETEND=false

LOCAL_URL=http://guia-apir.com
```

S'ha de crear la taula guia-apir a MYSQL.

La funcionalitat de mail no funcionarà si no es configura un servidor smtp. És pot utilitzar Mailtrap o el servidor smtp de Google. Aquestes dues opcions son les més senzilles d'implementar, si no s'ha d'utilitzar la funció de mail aleshores es pot obviar aquest pas però aleshores s'ha de deshabilitar l'enviament del mail d'activació i activar la compta per defecte en el controlador d'Usuaris si es vol crear una nova compta a l'entorn local. Cal tenir en compte que la funcionalitat de recuperar contrasenya requereix la funcionalitat de mail.

Amb això ja tindríem la api corrent al servidor. Per a poder accedir cal modificar l'arxiu hosts de windows. A Windows 10 es troba situat a C:

```
Windows
System32
drivers
```

etc. Afegir la següent línia al final.

127.0.0.1 guia-apir.com

Ara ja podem fer crides a la API a la direcció `guia-apir.com/api`, per fer proves directament als endpoints de la Api es pot utilitzar Postman.

Finalment, cal instal·lar NodeJs per a poder desenvolupar en Angular. Una volta instal·lat, utilitzant la consola de comanda de Windows anar a la carpeta `guia-apir-app` i executar el següent:

```
npm install  
ng serve -o
```

La primera comanda instal·la les dependències necessàries d'Angular i la segona obre un navegador al servidor de desenvolupament d'Angular, que per defecte es troba a `localhost:4200`. Si per algun motiu no trobes la comanda `ng` caldria instal·lar Angular CLI amb la següent comanda:

```
npm install -g @angular/cli
```

Una volta fet això ja tan sols quedaria obtenir les dades guardades a la base de dades del servei d'hostatgeria web de la UB. Però per a poder accedir-hi cal utilitzar un dels ordinadors de la facultat.

Si no s'ha instal·lat una versió de XAMPP que tingui la mateixa versió de PHP que el servei d'hostatgeria web de la UB aleshores es necessari instal·lar aquesta versió per a poder baixar les dependències corresponents a aquesta versió de PHP. El més senzill es instal·lar una maquina virtual Linux i instal·lar aquí la versió correcta de PHP i Composer únicament per a baixar les dependències abans de fer la pujada al servidor de la UB.